ORACLE
University

Integrated Cloud Applications & Platform Services

Oracle Application Express Workshop I

Student Guide – Volume I

D79653GC30 | D106871

Learn more from Oracle University at **education.oracle.com**

ORACLE

Authors

Sayantani Ghosh

Moitreyee Hazarika

Technical Contributors and Reviewers

Suresh Rajan

Anjani Pothula

David Peake

Chaitanya Koratamaddi

Joe Greenwald

Editors

Adrita Biswas

Aju Kumar

Moushmi Mukherjee

Graphic Editors

Prakash Dharmalingam

Anne Elizabeth

Pushparaj Kundar

Publishers

Pavithran Adka

Veena Narasimhan

Giri Venugopal

3009092019

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
The U.S. Government’s rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

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

Contents

1 Course Overview

Objectives 1-2
Course Goals 1-3
Course Road Map 1-4
Course Environment 1-8
Workspace Details 1-9
Accessing the labs Directory 1-10
Meet Course Persona: Stella 1-11
Meet Course Persona: Steve 1-12
Introduction to the Demo Application: Project Tracking System 1-13
Project Tracking System: Basic Reports 1-14
Project Tracking System: Interactive Grid 1-16
Project Tracking System: Supporting Reports 1-17
Snapshot of the Project Tracking System 1-18
Introduction to the Practice Application: GlobalMart Management Tool 1-19
Additional Resources – Oracle Application Express 19.1 1-20
Summary 1-21

I Unit I Introduction: Getting Started with Application Express

Stella Expresses Her Concern: Driving Factor Behind PTS I-2
Unit 1 Road Map I-3

2 Oracle Application Express: Introduction

You Are Here in This Course 2-2
Objectives 2-3
Steve Decides to Explore Oracle Application Express 2-4
Lesson Agenda 2-5
What Is Oracle Application Express? 2-6
Why Use Oracle Application Express? 2-7
Types of Applications 2-8
Applications Developed by Using Oracle Application Express 2-9
High-Level Architecture 2-10
Types of Installations 2-12
Quiz 2-13
Lesson Agenda 2-14

What Is a Workspace? 2-15
What Is an Internal Workspace? 2-16
Defining Roles 2-17
Quiz 2-19
Lesson Agenda 2-20
Signing In to a Workspace 2-21
Creating a Developer User 2-22
Using Application Express in Dark Mode 2-23
Practice 2-1 Overview: Using Oracle Application Express as a Workspace Administrator 2-24
Workspace Home Page 2-25
What Is App Builder? The First Component of Oracle Application Express 2-26
Types of Applications 2-27
What Is SQL Workshop? The Second Component of Oracle Application Express 2-28
Accessing SQL Workshop 2-29
Running SQL Commands 2-30
Importing and Running a SQL Script 2-31
What Is Team Development? The Third Component of Oracle Application Express 2-32
What Is App Gallery? The Fourth Component of Oracle Application Express 2-33
Accessing the App Gallery 2-34
Installing a Productivity Application 2-35
Running an Installed Productivity Application 2-36
Configuring the Setup 2-37
Unlocking an Installed Productivity Application 2-38
Exporting an Application 2-39
Importing an Application 2-40
Sample Datasets 2-42
Installing a Sample Dataset 2-43
Quiz 2-45
Practice 2-2 Overview: Using Oracle Application Express as a Developer 2-47
Summary 2-48

3 Creating a Database Application
You Are Here in This Course 3-2
Objectives 3-3
Steve Thinks of Developing PTS for Different Interfaces 3-4
Lesson Agenda 3-5
Accessing App Builder 3-6
App Builder Home Page 3-7

Lesson Agenda 3-8
Components of a Database Application 3-9
Database Application User Interfaces 3-10
Database Application Home Page 3-11
Page Designer: The World’s Most Advanced IDE 3-12
Page Designer: Drag and Drop from Gallery 3-13
Page Definition: Overview 3-14
Different Views of a Page 3-15
Switching Between Pages and View Types 3-16
Quiz 3-17
Lesson Agenda 3-19
Create Application Wizard 3-20
Accessing the Create Application Wizard 3-21
Different Ways of Creating a Database Application 3-22
Creating Database Application 3-23
Project Tracking System Home Page 3-24
Creating a Database Application from a File 3-25
Running an Application 3-26
Using the Developer Toolbar 3-27
Practice 3 Overview: Creating Database Applications 3-28
Summary 3-29

4 Working with Reports
You Are Here in This Course 4-2
Objectives 4-3
Steve Designs Reports for Tracking Projects 4-4
Lesson Agenda 4-5
Types of Reports 4-6
Different Ways of Creating a Report 4-7
Quiz 4-8
Lesson Agenda 4-9
Selecting the Appropriate Report Type 4-10
When Is Classic Report the Best Option? 4-11
Creating Classic Report with Form 4-12
Creating Classic Reports with Form 4-13
Classic Report 4-14
Practice 4-1 Overview: Creating and Modifying a Classic Report 4-15
List View Reports 4-16
Creating a List View Report 4-17
Modifying a List View Report 4-19
Practice 4-2 Overview: Creating and Modifying a List View Report 4-20

Column Toggle Reports 4-21
Creating a Column Toggle Report 4-22
Practice 4-3 Overview: Creating a Column Toggle Report 4-24
Quiz 4-25
Summary 4-26

5 Working with Interactive Reports
You Are Here in This Course 5-2
Objectives 5-3
When Is an Interactive Report the Best Option? 5-4
Lesson Agenda 5-5
How to Create an Interactive Report 5-6
Creating an Interactive Report 5-7
Interactive Report 5-9
Interactive Report with Link to Single Row View 5-10
Practice 5-1 Overview: Creating and Manipulating an Interactive Report 5-11
Lesson Agenda 5-12
Accessing Report Attributes 5-13
Editing Report Attributes 5-14
Customizing the Search Bar and Actions Menu 5-15
Specifying the Download Formats 5-16
Icon and Detail Views 5-17
Using the Link Column 5-18
Modifying the Interactive Report Query 5-19
Practice 5-2 Overview: Customizing an Interactive Report 5-20
Quiz 5-21
Summary 5-22

6 Working with Interactive Grids
You Are Here in This Course 6-2
Objectives 6-3
Steve Explores the Interactive Grid 6-4
Lesson Agenda 6-5
What Is an Interactive Grid? 6-6
An Interactive Grid with Form 6-7
Interactive Grid Versus Interactive Report 6-8
Lesson Agenda 6-9
Ways to Create an Interactive Grid 6-10
Creating an Interactive Grid (without Form) 6-11
Interactive Grid (without Form) 6-13
Creating an Interactive Grid (with Form) 6-14

Creating an Interactive Grid 6-15
Interactive Grid (with Form) 6-16
Key Attributes of an Interactive Grid: Region Attribute 6-18
Key Attributes of an Interactive Grid: Report Attribute 6-19
Key Attributes of an Interactive Grid: Column Attribute 6-20
Lesson Agenda 6-21
Customization: Rendering the Interactive Grid Editable 6-22
Interactive Grid: Editable 6-23
Customization: Creating Column Links 6-24
Customization: Modifying the Form 6-27
Customization: Modifying Fields on the Form 6-28
Customization: Creating a List of Values on a Column 6-29
Practice 6 Overview: Creating and Manipulating an Interactive Grid 6-31
Quiz 6-32
Summary 6-33

I Unit I Summary: Getting Started with Application Express
Unit I Road Map I-2

II Unit II Introduction: Building User-Friendly Web Applications
Steve Explores Oracle Application Express Further II-2
You Are Here in This Course II-3

7 Working with Pages and Regions
You Are Here in This Course 7-2
Objectives 7-3
Steve Works with Pages and Regions 7-4
Lesson Agenda 7-5
What Is a Page? 7-6
Accessing Page Definition 7-7
Page Modes: Normal, Modal, and Nonmodal 7-8
Page Designer Window 7-9
Page Designer UI Elements 7-10
Page Designer Toolbar 7-11
The Left Pane of Page Designer 7-12
The Central Pane of Page Designer 7-13
The Central Pane of Page Designer: Gallery 7-14
The Central Pane of Page Designer: Messages 7-15
The Central Pane of Page Designer: Page Search and Help 7-16
The Right Pane of Page Designer: Property Editor 7-17
Page Designer UI Elements: Property Editor: Code Editor 7-18

Editing Page Attributes 7-19
Running a Page 7-20
Runtime Developer Toolbar 7-21
Quiz 7-22
Lesson Agenda 7-23
Region and Region Types 7-24
Adding a Region: From the Rendering Tree 7-25
Adding a Region: From the Gallery 7-26
Positioning the Region 7-28
Example: Creating a Region Type (Region Display Selector) 7-29
Example: Creating a Region Display Selector 7-30
Conditional Display of Regions 7-32
Editing a Region 7-34
Editing a Region at Runtime (using Quick Edit option) 7-35
Editing a Region at Runtime (using Live Template Options) 7-36
Specifying a Region Header and Footer 7-37
Copying a Region 7-38
Example: Copying Regions 7-39
Creating a Subregion 7-40
Practice 7-1 Overview: Creating and Modifying Pages and Regions 7-41
Lesson Agenda 7-42
Global Page 7-43
Creating a Global Page Region 7-44
Example: Creating a Global Page Region 7-45
Practice 7-2 Overview: Working with Global Pages 7-46
Viewing Pages of the Application 7-47
Creating a Page Group 7-48
Example: Creating a Page Group 7-49
Copying a Database Application Page 7-51
Deleting a Page 7-52
Quiz 7-53
Summary 7-55

8 Managing Forms
You Are Here in This Course 8-2
Objectives 8-3
Stella Has a New Requirement: Wants a Solution 8-4
Steve Plans Data Entry Through Forms 8-5
Lesson Agenda 8-6
Forms: Introduction 8-7
Types of Forms 8-8

Lesson Agenda 8-9
Accessing the Create Form Wizards 8-10
ROWID Versus Primary Key 8-11
Creating a Form 8-12
Example: Creating a Form 8-13
Example: Adding a Form as a Region 8-16
Creating a Report with Form 8-17
Example: Creating a Report with Form 8-18
Creating a List View with Form 8-21
Example: Creating List View with Form 8-22
Creating a Stacked Master Detail Form 8-23
Example: Creating a Stacked Master Detail Form 8-24
Example: Creating a Stacked Master Detail 8-25
Quiz 8-26
Practice 8-1 Overview: Creating a Report with Form 8-28
Practice 8-2 Overview: Creating a List View with Form 8-29
Practice 8-3 Overview: Creating a Master Detail Form (Side By Side) 8-30
Lesson Agenda 8-31
Linking a Report to a Form 8-32
Example: Linking a Report to a Form 8-33
Practice 8-4 Overview: Creating a Master Detail Form (Stacked) 8-35
Practice 8-5 Overview: Creating a Master Detail Form (Drill-Down) 8-36
Using Quick Edit 8-37
Reordering Items 8-38
Changing Item Display Type 8-39
Customizing Forms 8-40
Quiz 8-41
Summary 8-42

9 Adding Items and Buttons
You Are Here in This Course 9-2
Objectives 9-3
Steve Enhances User Interface—Adds Items and Buttons to Pages 9-4
Lesson Agenda 9-5
Items 9-6
What Are Page Items? 9-7
Types of Page Items 9-8
What Are Application Items? 9-9
Lesson Agenda 9-10
Creating a Page Item 9-11
Example: Creating a Date Picker Item 9-12

Editing an Item 9-13
Examples: Editing an Item 9-14
Creating Quick Picks 9-16
Adding Subtypes 9-17
Quiz 9-18
Lesson Agenda 9-19
What Is an LOV? 9-20
Accessing the Lists of Values Page 9-21
Creating a Static LOV 9-22
Example: Creating a Static LOV 9-23
Associating the Static LOV with an Item 9-24
Creating a Dynamic LOV 9-25
Example: Creating a Dynamic LOV 9-26
Associating the Dynamic LOV with a Select List Item 9-27
Cascading LOV 9-28
Example: Creating a Cascading LOV 9-29
Quiz 9-31
Lesson Agenda 9-32
What Is a Button? 9-33
Creating a Button 9-34
Example: Creating a Button 9-35
Example: Editing Button Attributes 9-36
Modifying a Button to Redirect to a URL 9-37
Quiz 9-38
Practice 9-1 Overview: Adding Items and Buttons 9-39
Practice 9-2 Overview: Manipulating Items 9-40
Summary 9-41

10 Understanding Session State
You Are Here in This Course 10-2
Objectives 10-3
Steve Understands Session State 10-4
Lesson Agenda 10-5
What Is a Session State? 10-6
Session ID 10-7
Session Timeout 10-8
Setting Session Timeout 10-9
How Does Oracle Application Express Implement Session State? 10-10
Identifying the Parts of an Oracle Application Express URL 10-11
Quiz 10-13
Lesson Agenda 10-14

Viewing Session State 10-15
Referencing Session State 10-16
Example: Referencing Session State by Using Bind Variables 10-18
Example: Referencing Session State in Static Text 10-19
Clearing the Cache 10-20
Example: Clearing the Cache 10-21
Quiz 10-23
Practice 10-1 Overview: Understanding Session State 10-25
Summary 10-26

11 Including Page Processing
You Are Here in This Course 11-2
Objectives 11-3
Steve Includes Page Processing in the Application 11-4
Lesson Agenda 11-5
Page Rendering Versus Page Processing 11-6
Example: Page Rendering 11-7
Example: Page Processing 11-8
Types of Logic 11-10
Lesson Agenda 11-11
What Is a Computation? 11-12
Computation Examples 11-13
Creating Computations 11-14
Example: Creating Page-Processing Computations 11-15
Quiz 11-16
Lesson Agenda 11-17
What Is a Page Process? 11-18
Automatic Processing Processes 11-19
Reviewing an Automated Row Fetch Process 11-20
Reviewing an Automatic Row (DML) Processing Process 11-21
Creating an On Submit Process 11-22
Creating an On Load Process 11-23
Quiz 11-25
Lesson Agenda 11-26
What Are Validations? 11-27
Creating a Validation 11-28
Using the Create Validation Feature 11-29
Example: Creating an Item String Comparison Validation 11-30
Example: Creating a Regular Expression Validation 11-31
Example: Creating a SQL Validation 11-32
Example: Creating a PL/SQL Validation 11-33

Quiz 11-34
Lesson Agenda 11-35
What Is Branching? 11-36
Example: Creating a Branch 11-37
Practice 11-1 Overview: Creating and Manipulating Computations, Processes, and Validations 11-40
Summary 11-41

12 Using Dynamic Actions and Plug-Ins
You Are Here in This Course 12-2
Objectives 12-3
Steve Uses Dynamic Actions and Plug-ins 12-4
Lesson Agenda 12-5
Examples: Creating and Using Dynamic Actions 12-6
What Is a Dynamic Action? 12-7
Example 1: Enabling and Disabling an Item 12-8
General Steps to Create a Dynamic Action 12-9
Example 2: Setting the Value of an Item When Another Item Changes 12-11
Example 3: Clearing All Items on the Click of a Button 12-14
Example 3: Clearing All Items When a Button Is Clicked 12-15
Example 3: Clearing All Items on the Click of a Button 12-16
Example 4: Refreshing the Data in a Report Using Custom Filters 12-17
Example 4: Refreshing the Data in a Report Using Custom Filters: Overview 12-18
Example 4: Refreshing the Data in a Report Using Custom Filters 12-19
Example: Disabling a Button and Submitting the Page When Clicking a Button 12-21
Quiz 12-23
Practice 12-1 Overview: Creating and Using Dynamic Actions 12-25
Lesson Agenda 12-26
What Is a Plug-In? 12-27
Steps to Use a Plug-in in Your Application 12-28
Accessing the Plug-in Repository 12-29
Importing a Plug-In 12-31
Installing a Plug-In 12-32
Reviewing a Plug-in Definition 12-33
Using an Item Plug-in on a Page 12-35
Quiz 12-37
Additional Plug-in Examples 12-38
Adding the Modern Star Rating Item 12-39
Adding the Modern Star Rating Plug-in—Add Column 12-40
Adding the Modern Star Rating Item—Import and Install the Plug-in 12-41

Adding the Modern Star Rating Item Plug-in to a Form 12-42
Practice12-2 Overview: Importing and Using Plug-Ins 12-43
Quiz 12-44
Summary 12-45

13 Validating and Debugging Your Application
You Are Here in This Course 13-2
Objectives 13-3
Steve Validates and Debugs the Application 13-4
Lesson Agenda 13-5
Using the Advisor 13-6
Resolving Advisor Errors/Warnings 13-7
Practice 13-1 Overview: Using the Advisor 13-8
Lesson Agenda 13-9
Using User Interface Defaults 13-10
Creating User Interface Defaults for a Table 13-11
Managing Your Table Dictionary 13-12
Managing Your Attribute Dictionary 13-13
Reviewing Items and Report Columns 13-14
Modifying Attributes in the Dictionary 13-15
Quiz 13-16
Practice 13-2 Overview: Modifying the Attribute Dictionary 13-19
Lesson Agenda 13-20
What Is the Debug Option? 13-21
Enabling and Disabling Debug Mode 13-22
Debugging an Application 13-23
Viewing Debug Reports 13-24
Viewing Debug Reports (at Runtime) 13-25
Quiz 13-26
Summary 13-27

II Unit II Summary: Building User-Friendly Web Applications
Unit II Road Map II-2

III Unit III Introduction: Customizing Your Web Application
Steve Designs Application Navigation III-2
Unit III Road Map III-3

14 Adding Shared Components That Aid Navigation
You Are Here in This Course 14-2
Steve Builds a Navigation System into PTS 14-3

Objectives 14-4
Lesson Agenda 14-5
Shared Components: Examples 14-6
What Are Shared Components? 14-7
Lesson Agenda 14-8
Accessing Navigation Menu Page 14-9
Creating Navigation Menu Entries 14-10
Lesson Agenda 14-11
Lists 14-12
Accessing the Lists Page 14-13
Creating a Static List 14-14
Creating a Static List Region 14-15
Creating a Dynamic List 14-16
Creating a Dynamic List Region 14-18
Creating a List Region on Global Page 14-19
Lesson Agenda 14-20
Viewing a Breadcrumb 14-21
Creating Breadcrumb Entries 14-22
Reparenting Breadcrumbs 14-24
Lesson Agenda 14-25
Accessing the Navigation Bar Entries Page 14-26
Creating a Help Page 14-27
Creating a Navigation Bar Entry 14-28
Quiz 14-29
Practice14 Overview: Adding Shared Components That Aid Navigation 14-30
Summary 14-31

15 Working with Themes, Templates, and Files
You Are Here in This Course 15-2
Objectives 15-3
Steve Works with Application User Interface 15-4
Lesson Agenda 15-5
What Is a Theme? 15-6
Accessing the Themes Page 15-7
Creating a New Theme 15-8
Creating a Copy of an Existing Theme 15-9
Editing a Theme 15-10
Switching Between Themes 15-11
Example: Switching Between Themes 15-12
Changing the Theme Style 15-13
Using Universal Theme and Theme Roller 15-14

Accessing the Theme Roller 15-15
Customizing Your Theme Style Using Theme Roller 15-16
Customizing Your Theme Style Using Theme Roller – Changing Header 15-17
Customizing Your Theme Style Using Theme Roller – Changing Title Bar Background 15-18
Customizing Your Theme Style Using Theme Roller – Updating Navigation 15-19
Saving Theme Style in Theme Roller 15-20
Quiz 15-21
Lesson Agenda 15-22
What Are Templates? 15-23
Types of Templates 15-24
Accessing the Templates Page 15-25
Replacing a Template 15-26
Example: Replacing a Template 15-27
Creating a Copy of an Existing Template 15-28
Editing a Template 15-29
Applying a Template 15-30
Example: Applying a Template 15-31
Changing Default Templates in a Theme 15-32
Example: Changing Default Templates in a Theme 15-33
Overriding Application Defaults at the Page Level 15-34
Using Substitution Strings in Templates 15-35
Lesson Agenda 15-36
Uploading a Cascading Style Sheet 15-37
Referencing a Cascading Style Sheet 15-38
Uploading an Image 15-39
Example: Using an Uploaded Image 15-40
Quiz 15-42
Practice 15 Overview: Working with Themes, Templates, and Files 15-43
Summary 15-44

16 Implementing Security
Steve Implements Security in the Application 16-2
You Are Here in This Course 16-3
Objectives 16-4
Lesson Agenda 16-5
Securing an Application: Overview 16-6
Accessing Security Tasks 16-7
Lesson Agenda 16-8
Using Authentication 16-9
Accessing Authentication Schemes Page 16-10

Using Preconfigured Authentication Schemes 16-11
Examples: Preconfigured Authentication Schemes 16-12
Example: Creating Authentication Based on Preconfigured Schemes 16-13
Copying an Authentication Scheme 16-14
Quiz 16-15
Practice 16-1 Overview: Creating an Authentication Scheme 16-16
Lesson Agenda 16-17
Using Authorization 16-18
Where Can You Implement Authorization? 16-19
Methods to Implement Authorization 16-20
Creating an Authorization Scheme from the Scratch 16-21
Creating Users with Roles – Example: Create User 1 16-23
Creating Users with Roles – Example: Create User 2 16-24
Creating an Access Control Page 16-25
Setting Access Control Mode 16-26
Creating Access Control List 16-27
Applying an Authorization Scheme to an Application 16-29
Applying an Authorization Scheme to a Page 16-30
Applying an Authorization Scheme to a Column in a Report 16-31
Example: Applying an Authorization Scheme to a Column in a Report 16-32
Quiz 16-33
Practice 16-2 Overview: Restricting Users by Using Access Control 16-34
Lesson Agenda 16-35
Session State Protection 16-36
Enabling Session State Protection from the Edit Application Page 16-37
Enabling Session State Protection from the Session State Protection Page 16-38
Configuring Session State Protection 16-39
Configuring Session State Protection by Using a Wizard 16-40
Configuring Session State Protection for Pages and Items 16-41
Configuring Session State Protection for Application Items 16-42
Practice 16-3 Overview: Enabling Session State Protection 16-43
Summary 16-44

17 Managing Application Navigation
Steve Tries to Improve Application Navigation 17-2
You Are Here in This Course 17-3
Objectives 17-4
Lesson Agenda 17-5
Building a Hierarchical List with Images 17-6
Building a Hierarchical List 17-8
Example: Building a Hierarchical List 17-9

Building a Hierarchical List with Images 17-10
Practice 17-1 Overview: Building a Hierarchical List with Images 17-13
Lesson Agenda 17-14
Building a Database-Driven Navigation Report 17-15
Quiz 17-19
Practice17-2 Overview: Building a Database-Driven Report 17-20
Lesson Agenda 17-21
Building a Site Map 17-22
Adding Site Map as a Navigation Bar Entry 17-27
Example: Adding Site Map as a Navigation Bar Entry 17-28
Quiz 17-29
Practice17-3 Overview: Building a Site Map 17-30
Lesson Agenda 17-31
Enforcing Authorization on Your Site Map 17-32
Practice 17-4 Overview: Enforcing Authorization on the Site Map 17-35
Summary 17-36

III Unit III Summary: Customizing Your Web Application
Unit III Road Map III-2

IV Unit IV Introduction: Enhancing Your Web Application
Steve Enhances the Project Tracking System Application IV-2
Unit IV Road Map IV-3

18 Extending Your Application
Steve Extends the PTS Application 18-2
You Are Here in This Course 18-3
Objectives 18-4
Lesson Agenda 18-5
Data Load Wizard 18-6
Creating Data Load Wizard Pages 18-7
Data Load Wizard Pages 18-9
Practice 18-1: Creating Data Load Wizard Pages 18-10
Lesson Agenda 18-11
Contact Us Page 18-12
Creating a Send E-Mail Process 18-14
Lesson Agenda 18-15
Creating an Upload and Download Page 18-16
Adding a File Browse Item on the Upload Download Page 18-17
Practice 18-2: Adding an Upload and Download Page 18-18
Summary 18-19

19 Creating and Editing Charts

Introducing Visual Aids for Representing Data 19-2
You Are Here in This Course 19-3
Objectives 19-4
Lesson Agenda 19-5
Search a Workspace for AnyChart Using SQL Commands 19-6
Upgrading from AnyCharts to JET Charts 19-8
Lesson Agenda 19-9
Create Charts 19-10
Creating SQL Queries for Charts 19-11
Creating a Bar Chart 19-12
View and Edit Chart Attributes 19-14
Lesson Agenda 19-15
View and Edit Chart Attributes 19-16
Visualizing Data in a Line with Area Chart and Pie Chart 19-17
Visualizing Data in a Line with Area Chart 19-18
Visualizing Data in a Pie Chart 19-19
Practice 19-1 Overview: Creating and Editing Charts 19-20
Practice19-2 Overview: Creating a Pie and a Donut Chart 19-21
Lesson Agenda 19-22
Create a Pyramid Chart 19-23
Create a Status Meter Gauge Chart 19-26
Viewing and Analyzing a Combined Chart 19-27
Practice19-3 Overview: Enhanced Charting Examples 19-28
Quiz 19-29
Summary 19-30

20 Adding Calendars and Trees

Explores Tools to Organize Project Activities 20-2
You Are Here in This Course 20-3
Objectives 20-4
Lesson Agenda 20-5
Creating a Calendar 20-6
Creating a Calendar on Home Page 20-7
Editing Calendar Attributes 20-8
Dragging and Dropping Calendar Entries 20-9
Defining Links in a Calendar 20-10
Linking to the Calendar from a Button 20-11
Practice 20-1 Overview: Creating a Calendar 20-13
Lesson Agenda 20-14

What Is a Tree?   20-15
Creating a Tree   20-16
Exploring a Tree   20-21
Practice 20-2 Overview: Creating a Tree Whose Nodes Link to a Different Page   20-23
Summary   20-24

21 Managing Application Feedback

Introducing Feedback Mechanism   21-2
You Are Here in This Course   21-3
Objectives   21-4
Lesson Agenda   21-5
Managing Application Feedback   21-6
Lesson Agenda   21-7
What Is Team Development?   21-8
Tracking the Progress of Your Application Development Project   21-9
Creating Features   21-10
Creating Milestones   21-11
Creating Bugs   21-12
Creating To Dos   21-13
Quiz   21-14
Lesson Agenda   21-17
Review the Progress of Your Milestones and Features   21-18
Lesson Agenda   21-19
Enabling Feedback in your Application   21-20
Step 1: Enabling Feedback in Application Properties   21-21
Step 2: Creating a Feedback Page   21-22
Step 3: Submitting Feedback   21-23
Step 4: Accessing Feedback in Team Development   21-24
Step 4.1: Accessing Feedback as an End User   21-25
Quiz   21-26
Practice 21 Overview: Adding and Monitoring Feedback in Your Application   21-28
Summary   21-29

IV Unit IV Summary: Enhancing Your Web Application

Unit IV Road Map   IV-2

1

Course Overview

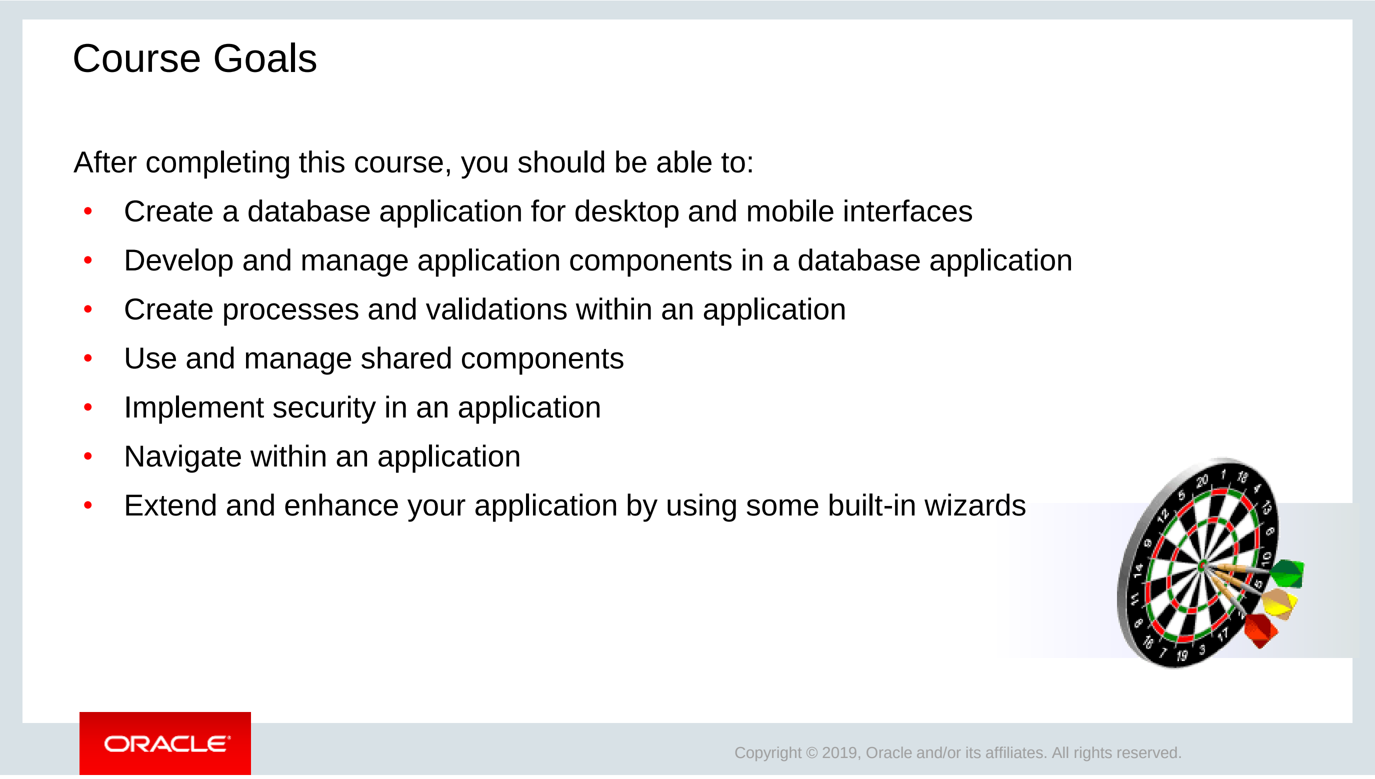
Objectives

After completing this lesson, you should be able to describe the following:

- Course goals
- Course road map
- Course environment
- Course persona: Stella and Steve
- Project Tracking System (PTS): Demo Application
- GlobalMart Management Tool (GMT): Practice Application

ORACLE

This course is developed based on the assumption that the target audience has a general understanding of relational database concepts on which they will run Oracle Application Express 19.1.



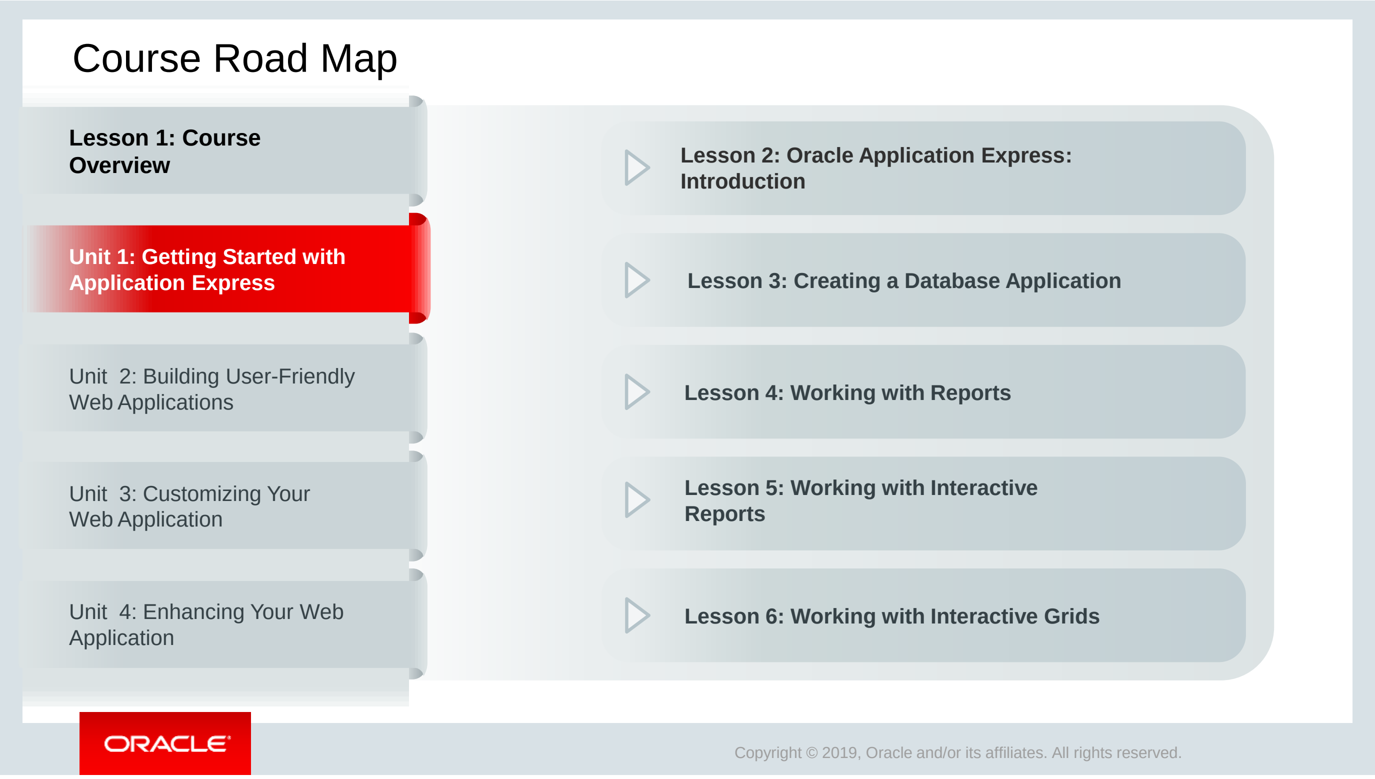
Course Goals

After completing this course, you should be able to:

- Create a database application for desktop and mobile interfaces
- Develop and manage application components in a database application
- Create processes and validations within an application
- Use and manage shared components
- Implement security in an application
- Navigate within an application
- Extend and enhance your application by using some built-in wizards

This course introduces you to Oracle Application Express, a rapid web application development tool for Oracle Database. In this course, you learn:

- About the features and benefits of Oracle Application Express
- How to use and manage components to build a complete and secure web application
- How to implement security in your application
- How to customize your application



Course Road Map

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

Lesson 2: Oracle Application Express: Introduction

Lesson 3: Creating a Database Application

Lesson 4: Working with Reports

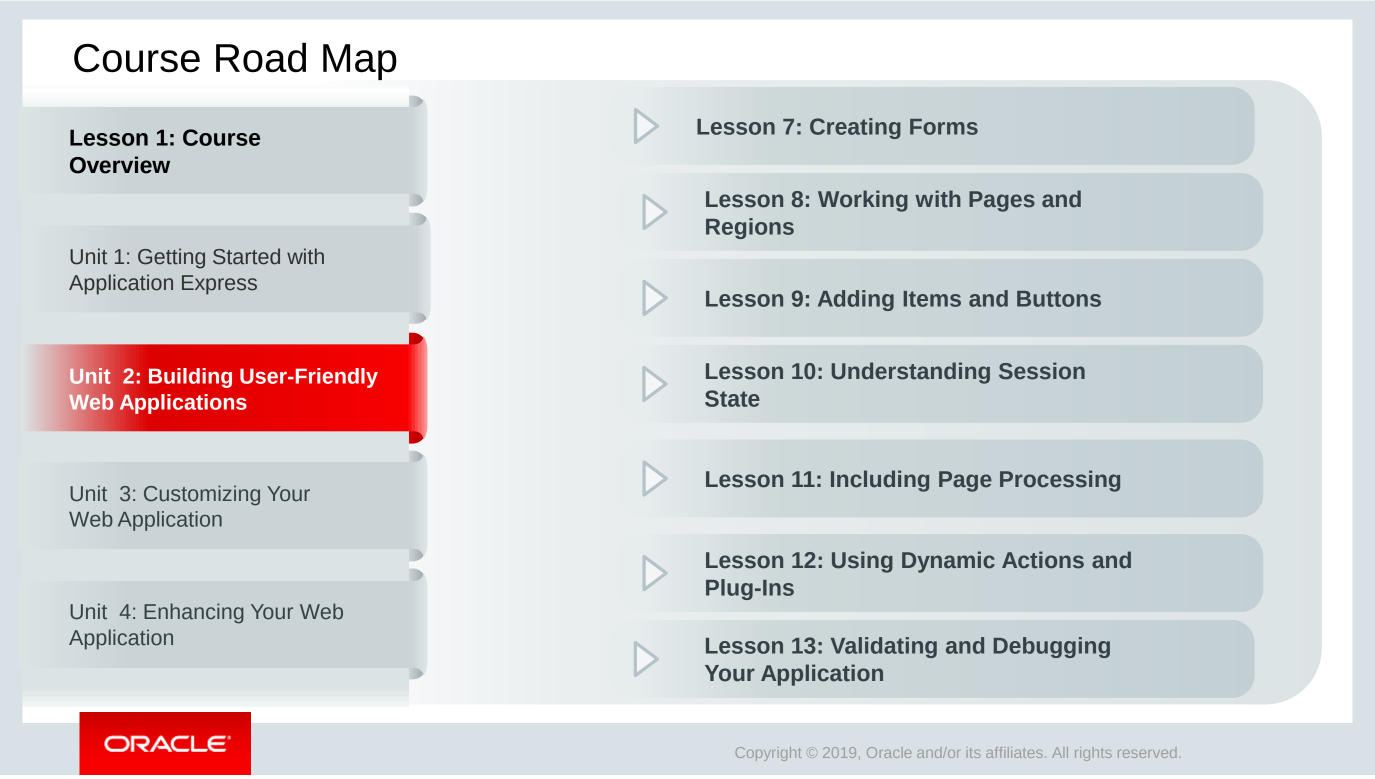
Lesson 5: Working with Interactive Reports

Lesson 6: Working with Interactive Grids

This slide is a graphical depiction of the overall course structure, particularly unit 1.

In unit 1, you learn how to create:

- Database applications using Oracle Application Express 19.1
- Classic reports
- Interactive reports
- Interactive grids



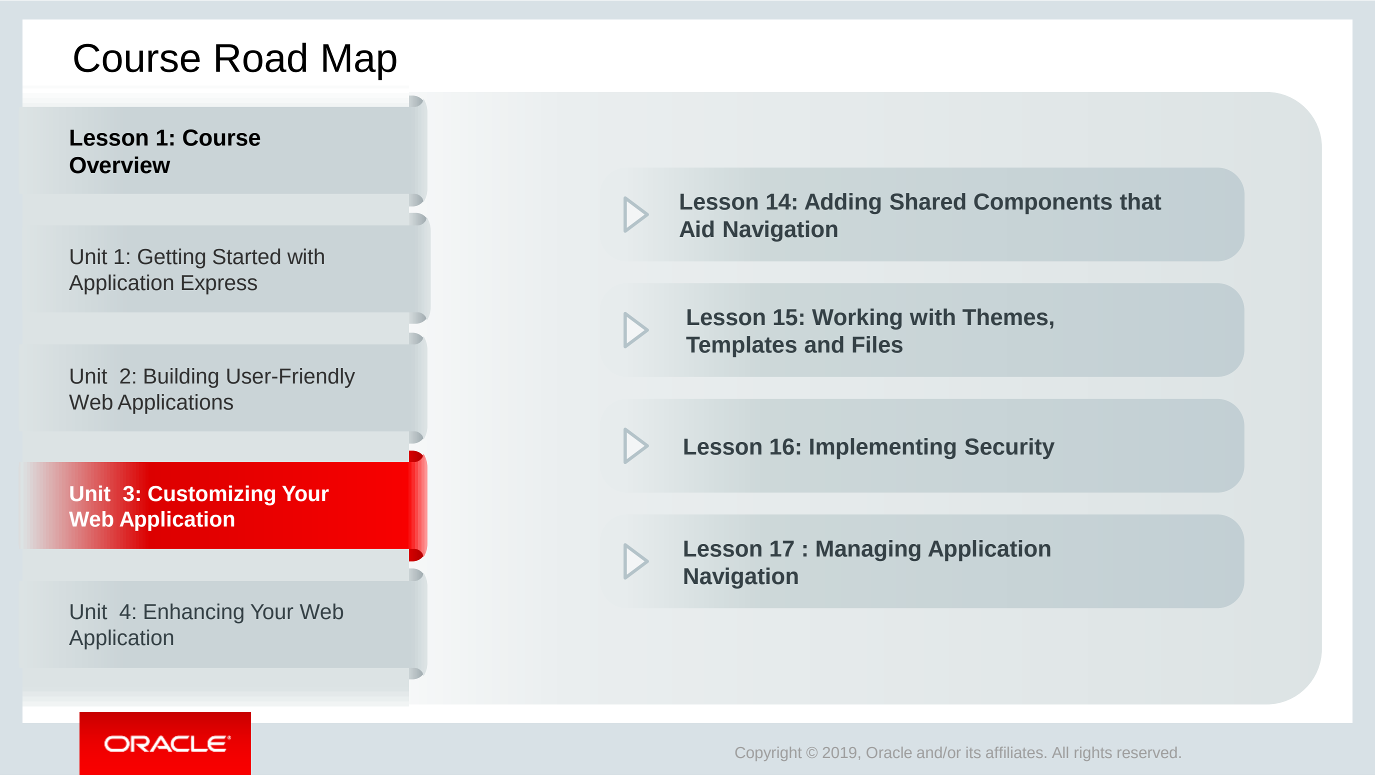
Course Road Map

- Lesson 1: Course Overview
- Unit 1: Getting Started with Application Express
- Unit 2: Building User-Friendly Web Applications
- Unit 3: Customizing Your Web Application
- Unit 4: Enhancing Your Web Application

- Lesson 7: Creating Forms
- Lesson 8: Working with Pages and Regions
- Lesson 9: Adding Items and Buttons
- Lesson 10: Understanding Session State
- Lesson 11: Including Page Processing
- Lesson 12: Using Dynamic Actions and Plug-Ins
- Lesson 13: Validating and Debugging Your Application

In Unit 2, you learn how to:

- Build a user-friendly interface to your application with the help of forms, pages, and regions
- Add items, buttons, and processing to your pages to support validations and computations
- Use dynamic actions and plug-ins in your application
- Validate and debug your application



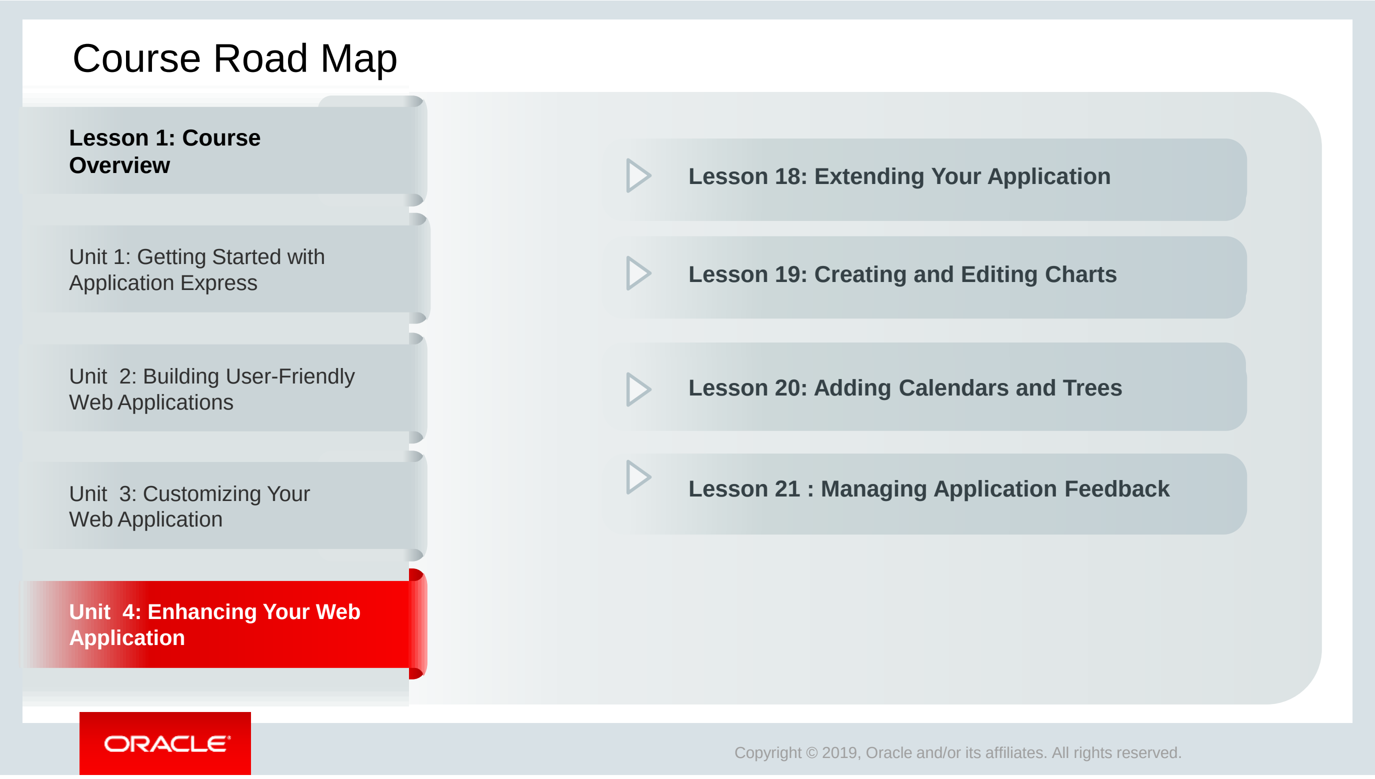
Course Road Map

- **Lesson 1: Course Overview**
- Unit 1: Getting Started with Application Express
- Unit 2: Building User-Friendly Web Applications
- **Unit 3: Customizing Your Web Application**
- Unit 4: Enhancing Your Web Application

- **Lesson 14: Adding Shared Components that Aid Navigation**
- **Lesson 15: Working with Themes, Templates and Files**
- **Lesson 16: Implementing Security**
- **Lesson 17 : Managing Application Navigation**

In Unit 3, you learn how to:

- Include navigation in your application with the help of shared components
- Work with themes, templates, and files in your application
- Implement page-level authorization to make your application highly secure



Course Road Map

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

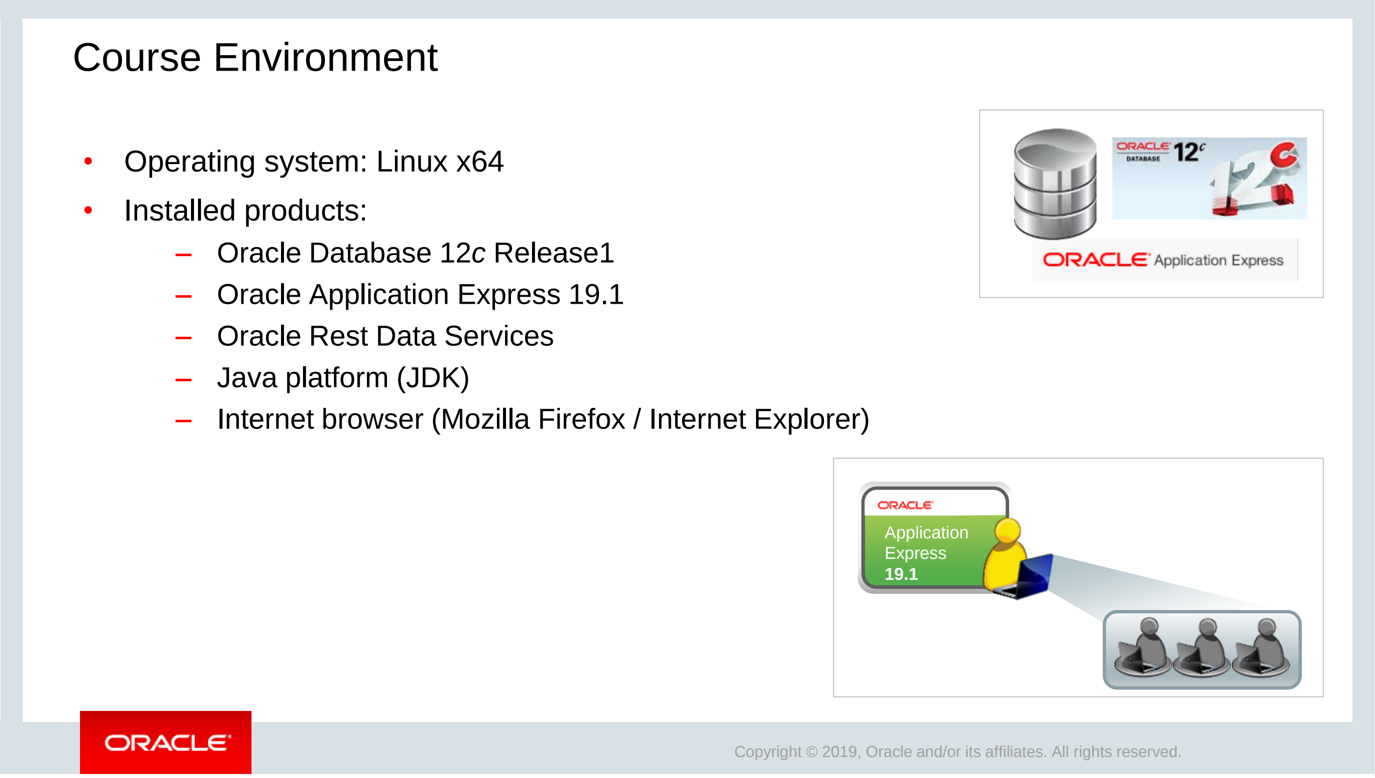
Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- **Lesson 18: Extending Your Application**
- **Lesson 19: Creating and Editing Charts**
- **Lesson 20: Adding Calendars and Trees**
- **Lesson 21 : Managing Application Feedback**

In Unit 4, you learn how to:

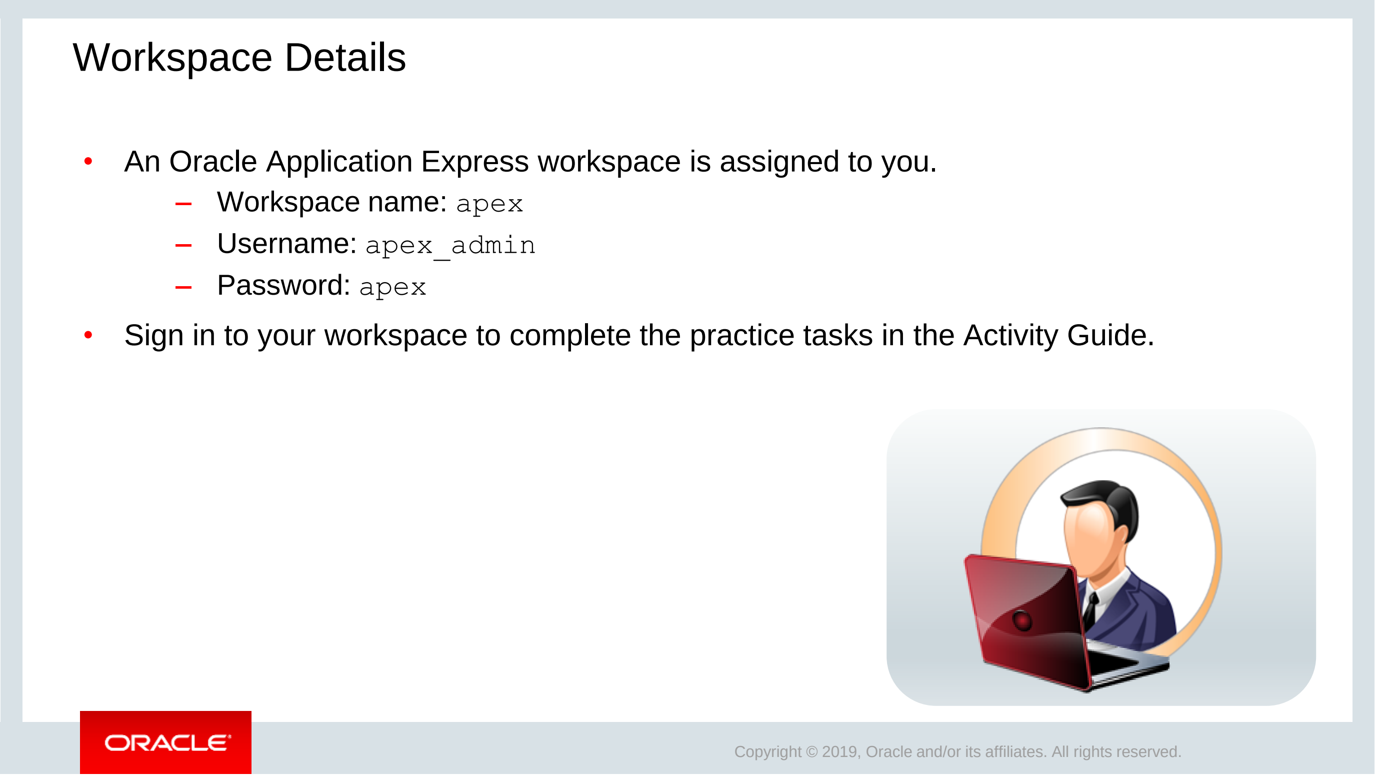
- Add advanced features in your application by creating charts, calendars, and trees
- Use the printing feature in Oracle Application Express
- Manage feedback in your application



Course Environment

- Operating system: Linux x64
- Installed products:
  - Oracle Database 12*c* Release1
  - Oracle Application Express 19.1
  - Oracle Rest Data Services
  - Java platform (JDK)
  - Internet browser (Mozilla Firefox / Internet Explorer)

The classroom setup uses a client/server architecture. The instructor machine is set up with Oracle Database and the required software to run Oracle Application Express. You will sign in to the student machine that is assigned to you. From the student machine, you access the Oracle Application Express workspace that is assigned to you by using a web browser.



Workspace Details

- An Oracle Application Express workspace is assigned to you.
  - Workspace name: `apex`
  - Username: `apex_admin`
  - Password: `apex`
- Sign in to your workspace to complete the practice tasks in the Activity Guide.

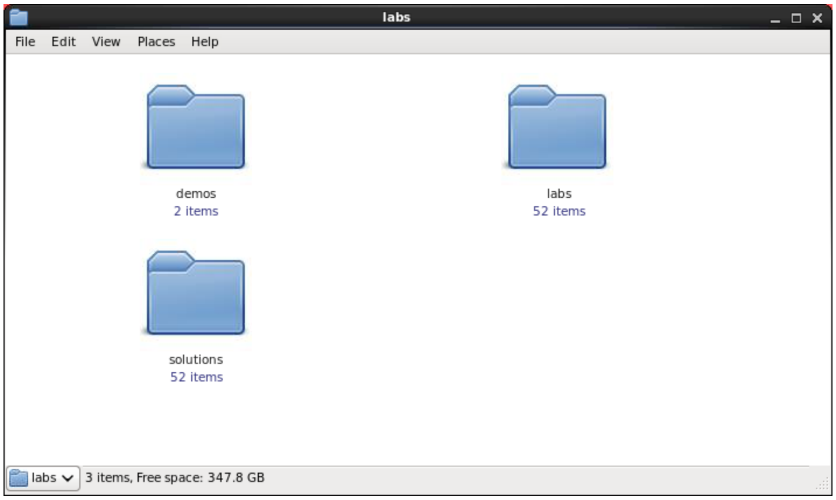
Your instructor assigns a workspace to you. The workspace name, username, and password are listed in the slide. You need to sign in to this workspace to complete all the practices in the Activity Guide for this course.

To access the Oracle Application Express development instance, open a web browser and enter the following URL in the address bar:

`http://<hostname>:8080/apex`

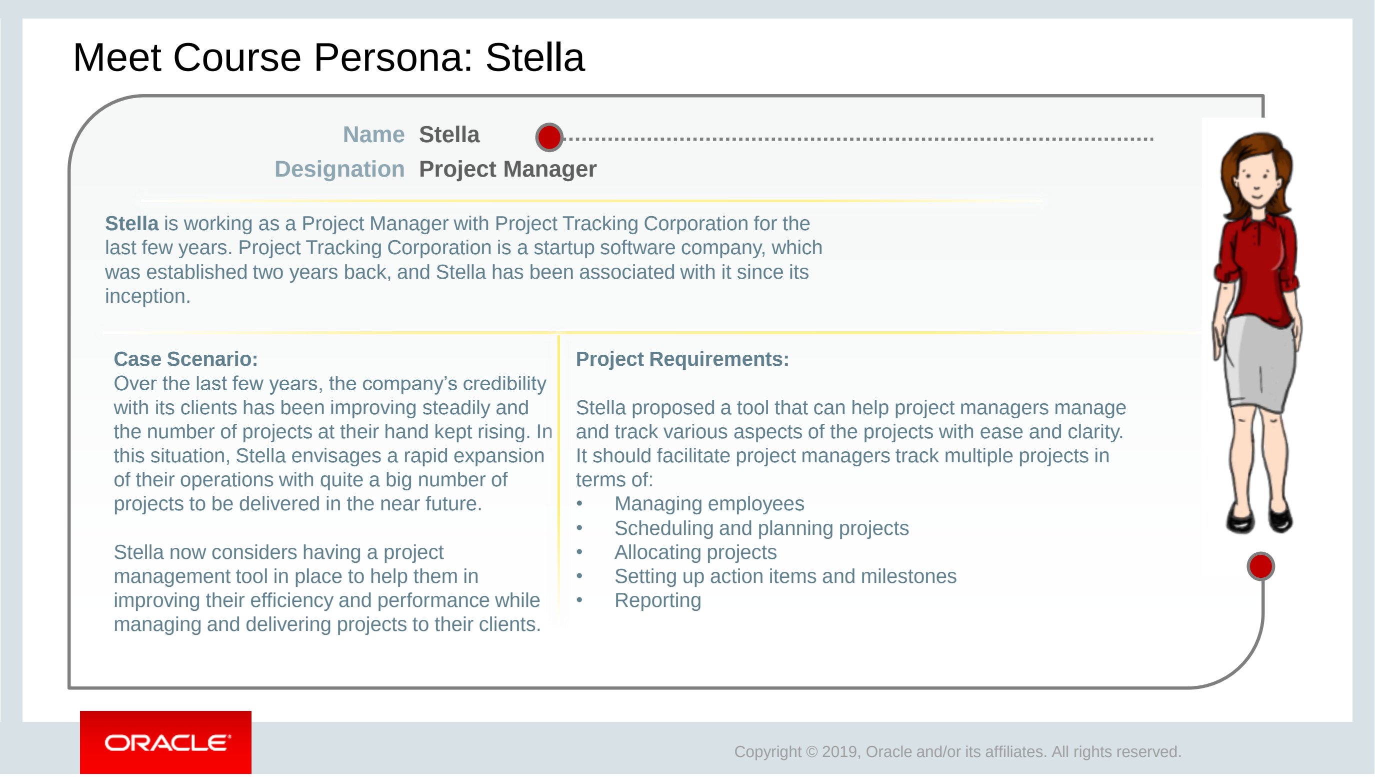
Note: The `hostname` is the IP address of the instructor machine.

Accessing the `labs` Directory



All the files that are required to complete the practices are available in the `labs` directory.
To access the `labs` directory, from the Applications menu, select System Tools > File Browser. From the `oracle` directory, open the `labs` directory. You will see four folders:

- **labs:** All the files required to perform the practices given in the Activity Guide. You can use this location to save the files while performing the practices, if required.
- **solutions:** The solution scripts for the practices given in the Activity Guide. They can be used as the catch-up applications that you can import in case you were not able to complete any practice.
- **demos/files:** All the files required to perform the demonstrations referenced in the lessons
- **demos/catchup:** Catch-up applications that the instructor can use



Meet Course Persona: Stella

Name Stella

Designation Project Manager

Stella is working as a Project Manager with Project Tracking Corporation for the last few years. Project Tracking Corporation is a startup software company, which was established two years back, and Stella has been associated with it since its inception.

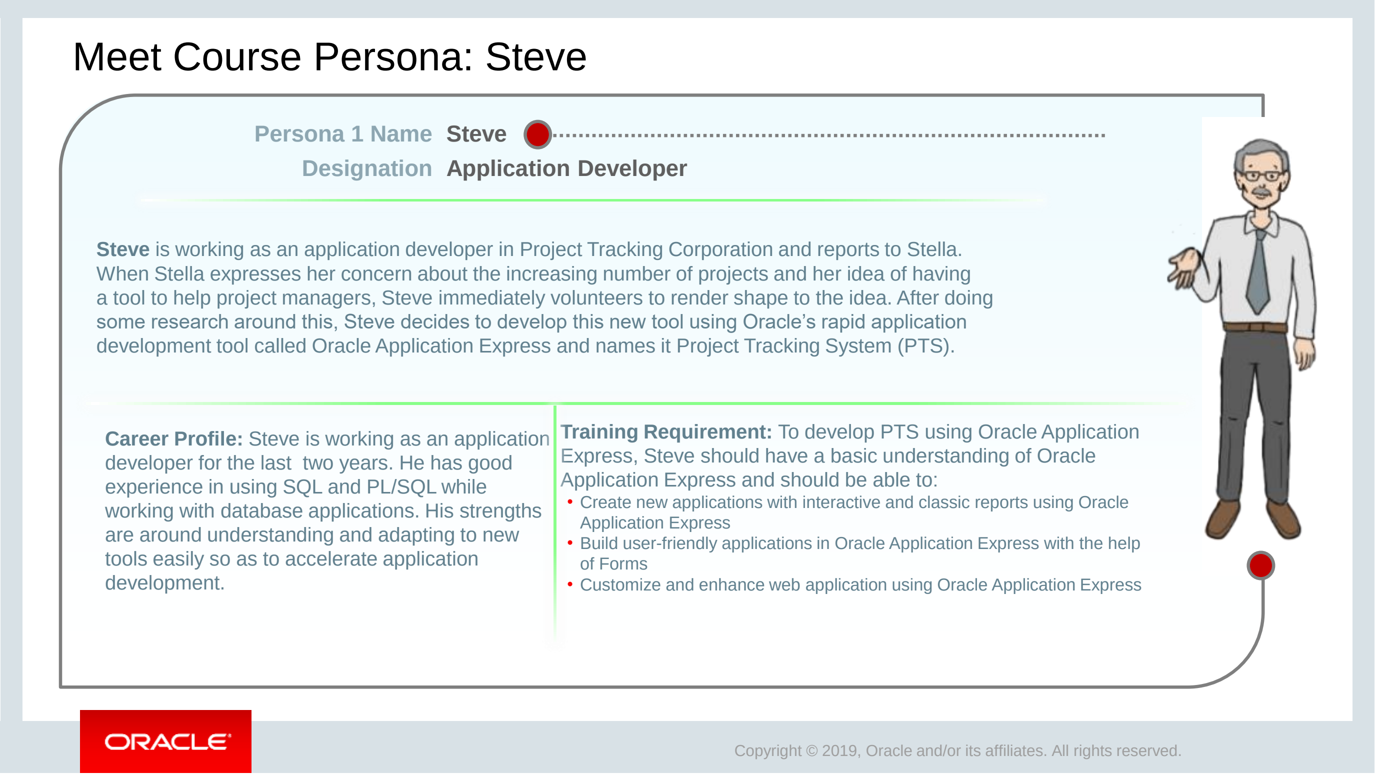
Case Scenario:
Over the last few years, the company's credibility with its clients has been improving steadily and the number of projects at their hand kept rising. In this situation, Stella envisages a rapid expansion of their operations with quite a big number of projects to be delivered in the near future.

Stella now considers having a project management tool in place to help them in improving their efficiency and performance while managing and delivering projects to their clients.

Project Requirements:

Stella proposed a tool that can help project managers manage and track various aspects of the projects with ease and clarity. It should facilitate project managers track multiple projects in terms of:

- Managing employees
- Scheduling and planning projects
- Allocating projects
- Setting up action items and milestones
- Reporting



Meet Course Persona: Steve

Persona 1 Name **Steve**

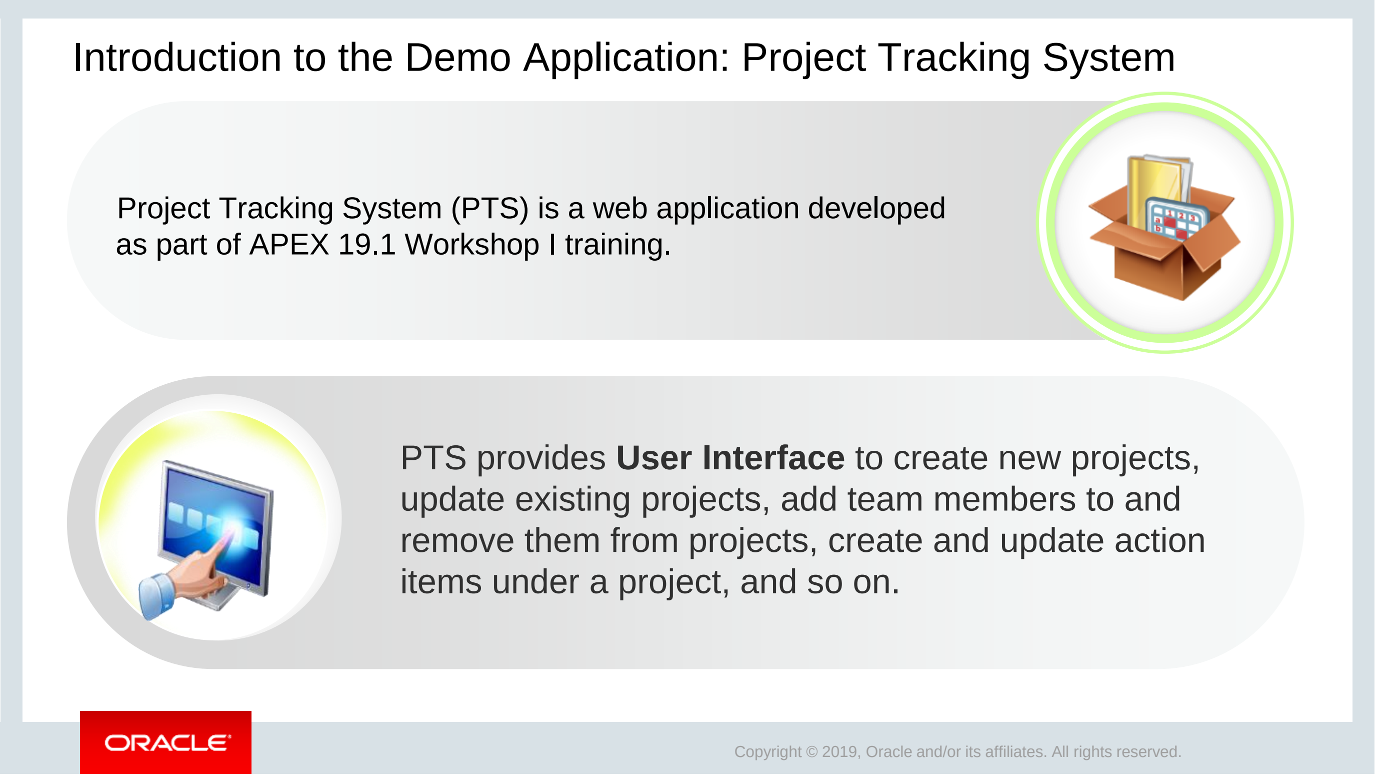
Designation **Application Developer**

Steve is working as an application developer in Project Tracking Corporation and reports to Stella. When Stella expresses her concern about the increasing number of projects and her idea of having a tool to help project managers, Steve immediately volunteers to render shape to the idea. After doing some research around this, Steve decides to develop this new tool using Oracle's rapid application development tool called Oracle Application Express and names it Project Tracking System (PTS).

Career Profile: Steve is working as an application developer for the last two years. He has good experience in using SQL and PL/SQL while working with database applications. His strengths are around understanding and adapting to new tools easily so as to accelerate application development.

Training Requirement: To develop PTS using Oracle Application Express, Steve should have a basic understanding of Oracle Application Express and should be able to:

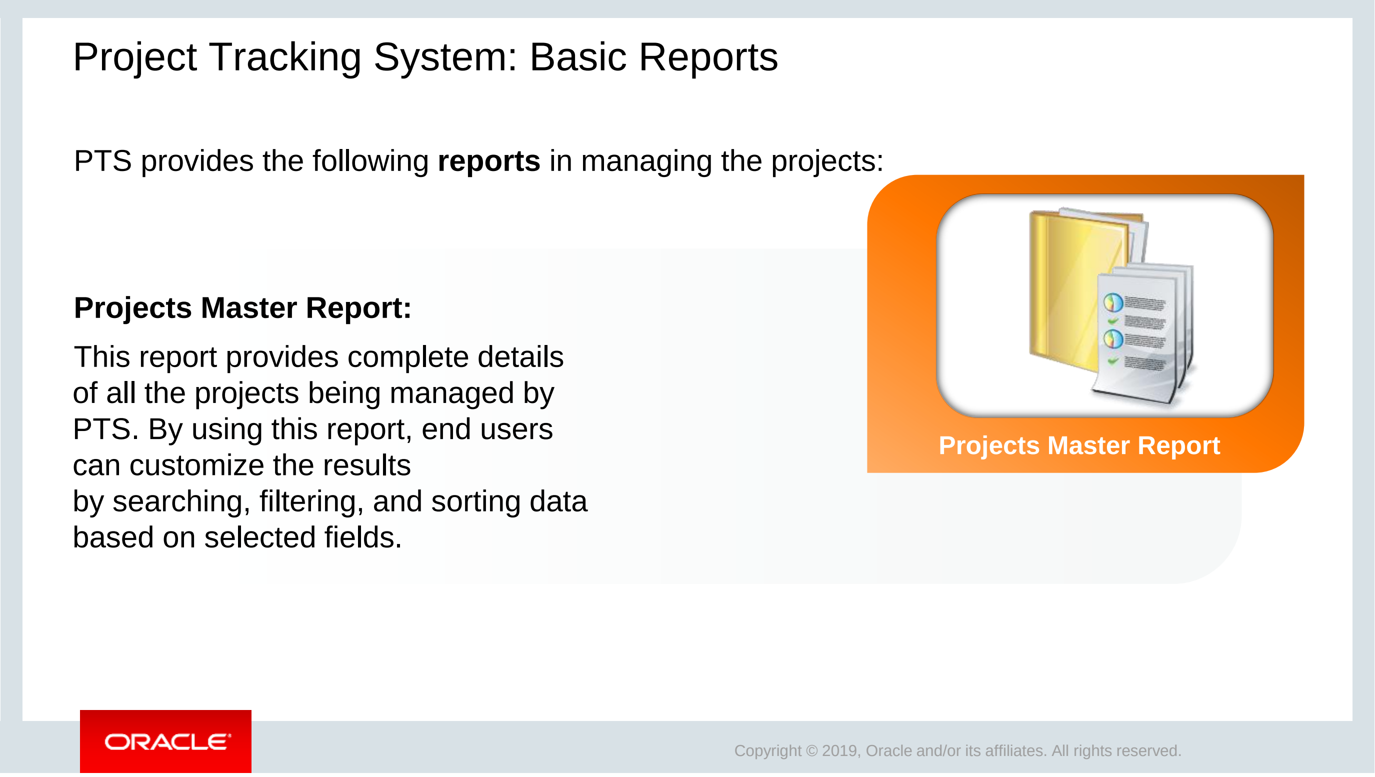
- Create new applications with interactive and classic reports using Oracle Application Express
- Build user-friendly applications in Oracle Application Express with the help of Forms
- Customize and enhance web application using Oracle Application Express



Introduction to the Demo Application: Project Tracking System

Project Tracking System (PTS) is a web application developed as part of APEX 19.1 Workshop I training.

PTS provides **User Interface** to create new projects, update existing projects, add team members to and remove them from projects, create and update action items under a project, and so on.



Project Tracking System: Basic Reports

PTS provides the following **reports** in managing the projects:

Projects Master Report:

This report provides complete details of all the projects being managed by PTS. By using this report, end users can customize the results by searching, filtering, and sorting data based on selected fields.

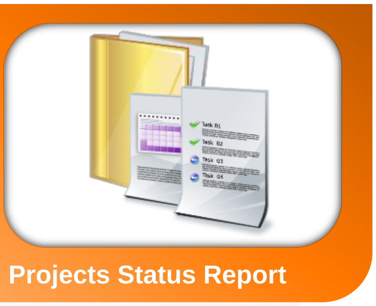
Projects Master Report

Project Tracking System: Basic Reports

Projects Status Report:

This report provides a list of completed and pending tasks of a project as of a given date. It will also give an insight into upcoming milestones for better tracking.

This will be a classic report without any end-user customization.



Projects Status Report

Project Tracking System: Interactive Grid

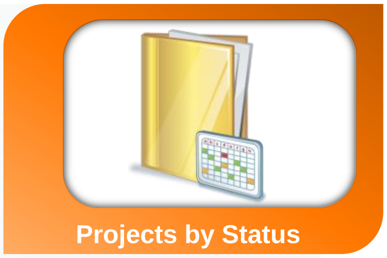
Projects Master Document – Interactive Grid:

This report will give a complete list of all documents, such as project plan, data modelling diagrams, schema tracking, technical and functional requirement documents, risk mitigation plan, SQL scripts, and so on that are used in all projects. By using this report, the end user, who is a project manager in this scenario, will be able to customize the results by searching, filtering, and sorting data based on selected projects.

Project Tracking System: Supporting Reports

Projects by Status:

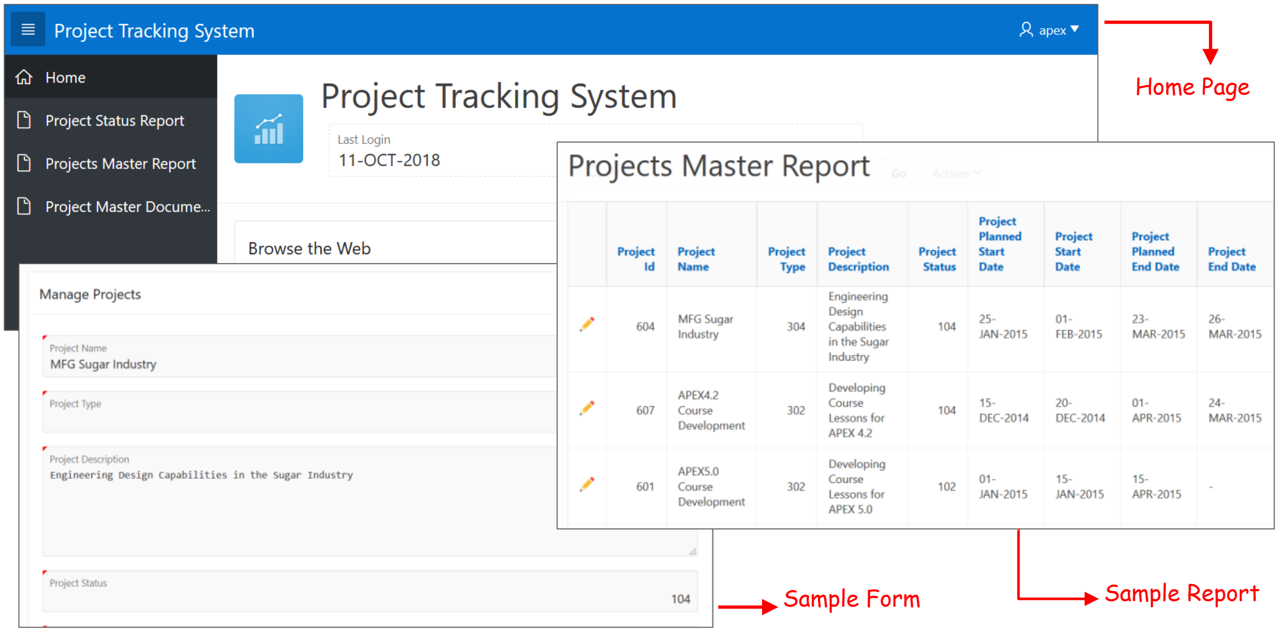
This report lists all the projects categorized by project status. This will be a user-customized report that uses Projects Master Report as its base.



Projects by Managers:

This report lists all the projects grouped by project managers who are managing those projects. This will be a user-customized report that uses Projects Master Report as its base.

Snapshot of the Project Tracking System



Project Tracking System

Last Login
11-OCT-2018

Manage Projects

Project Name
MFG Sugar Industry

Project Type

Project Description
Engineering Design Capabilities in the Sugar Industry

Project Status
104

Projects Master Report

	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Project End Date
✎	604	MFG Sugar Industry	304	Engineering Design Capabilities in the Sugar Industry	104	25-JAN-2015	01-FEB-2015	23-MAR-2015	26-MAR-2015
✎	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-2014	20-DEC-2014	01-APR-2015	24-MAR-2015
✎	601	APEX5.0 Course Development	302	Developing Course Lessons for APEX 5.0	102	01-JAN-2015	15-JAN-2015	15-APR-2015	-

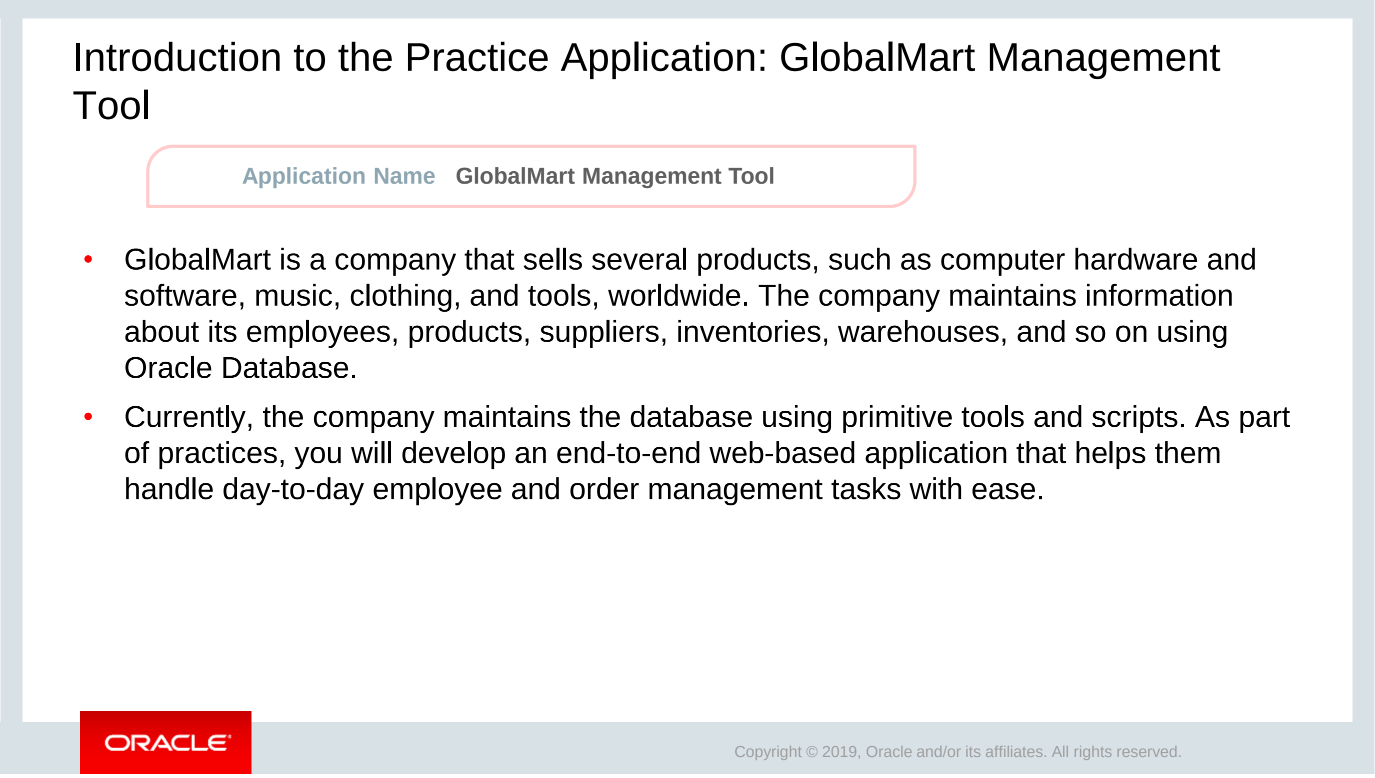
Home Page

Sample Form

Sample Report

In this course, you create the Project Tracking System (PTS) application as part of the demonstrations. PTS is a simple project management application that is designed to automate project management activities, such as maintaining project status and milestones, generating project reports, and so on.

Creating the PTS application will help you understand the concepts and features of Oracle Application Express and apply them while building the database application.

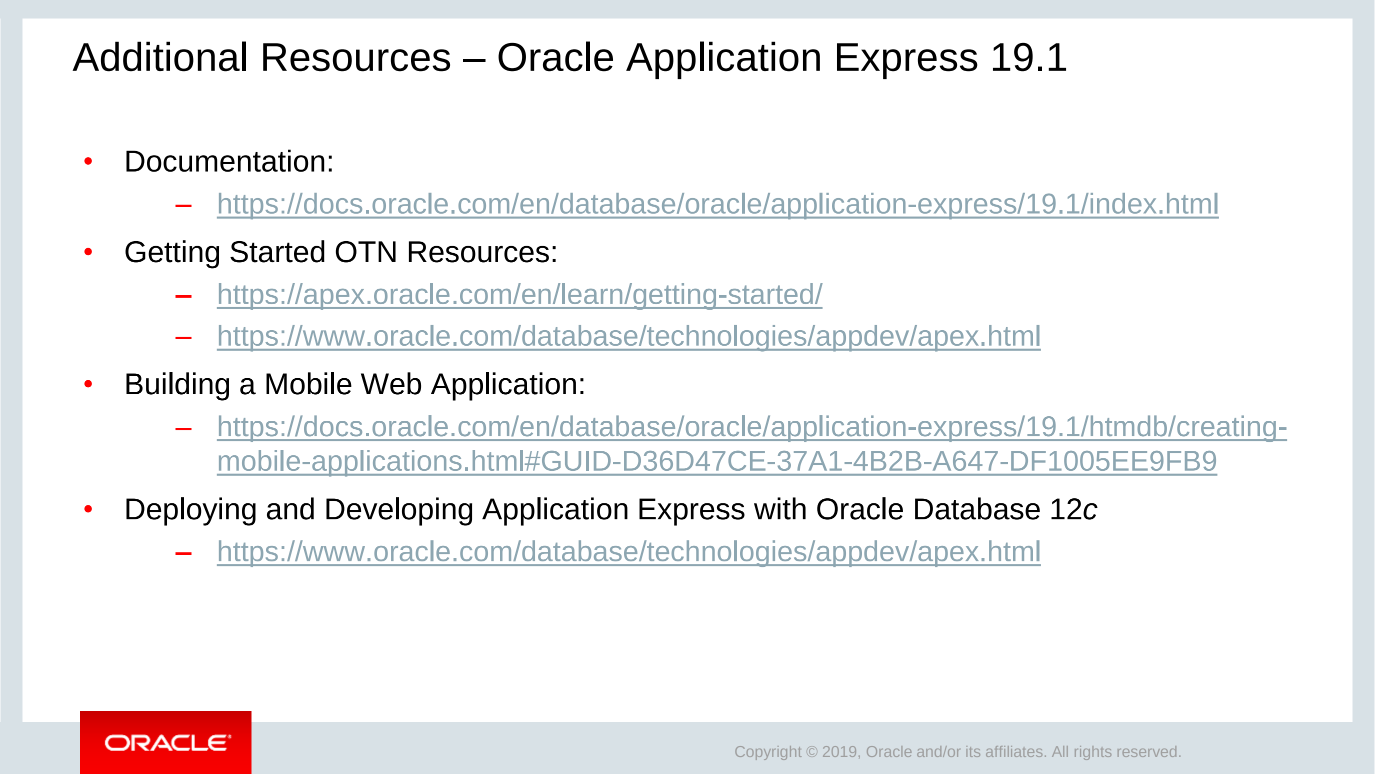


Introduction to the Practice Application: GlobalMart Management Tool

Application Name GlobalMart Management Tool

- GlobalMart is a company that sells several products, such as computer hardware and software, music, clothing, and tools, worldwide. The company maintains information about its employees, products, suppliers, inventories, warehouses, and so on using Oracle Database.
- Currently, the company maintains the database using primitive tools and scripts. As part of practices, you will develop an end-to-end web-based application that helps them handle day-to-day employee and order management tasks with ease.

As part of the practices, you will develop another application called the GlobalMart Management Tool (GMT), which is an Order Management application.



Additional Resources – Oracle Application Express 19.1

- Documentation:
  - https://docs.oracle.com/en/database/oracle/application-express/19.1/index.html
- Getting Started OTN Resources:
  - https://apex.oracle.com/en/learn/getting-started/
  - https://www.oracle.com/database/technologies/appdev/apex.html
- Building a Mobile Web Application:
  - https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/creating-mobile-applications.html#GUID-D36D47CE-37A1-4B2B-A647-DF1005EE9FB9
- Deploying and Developing Application Express with Oracle Database 12*c*
  - https://www.oracle.com/database/technologies/appdev/apex.html

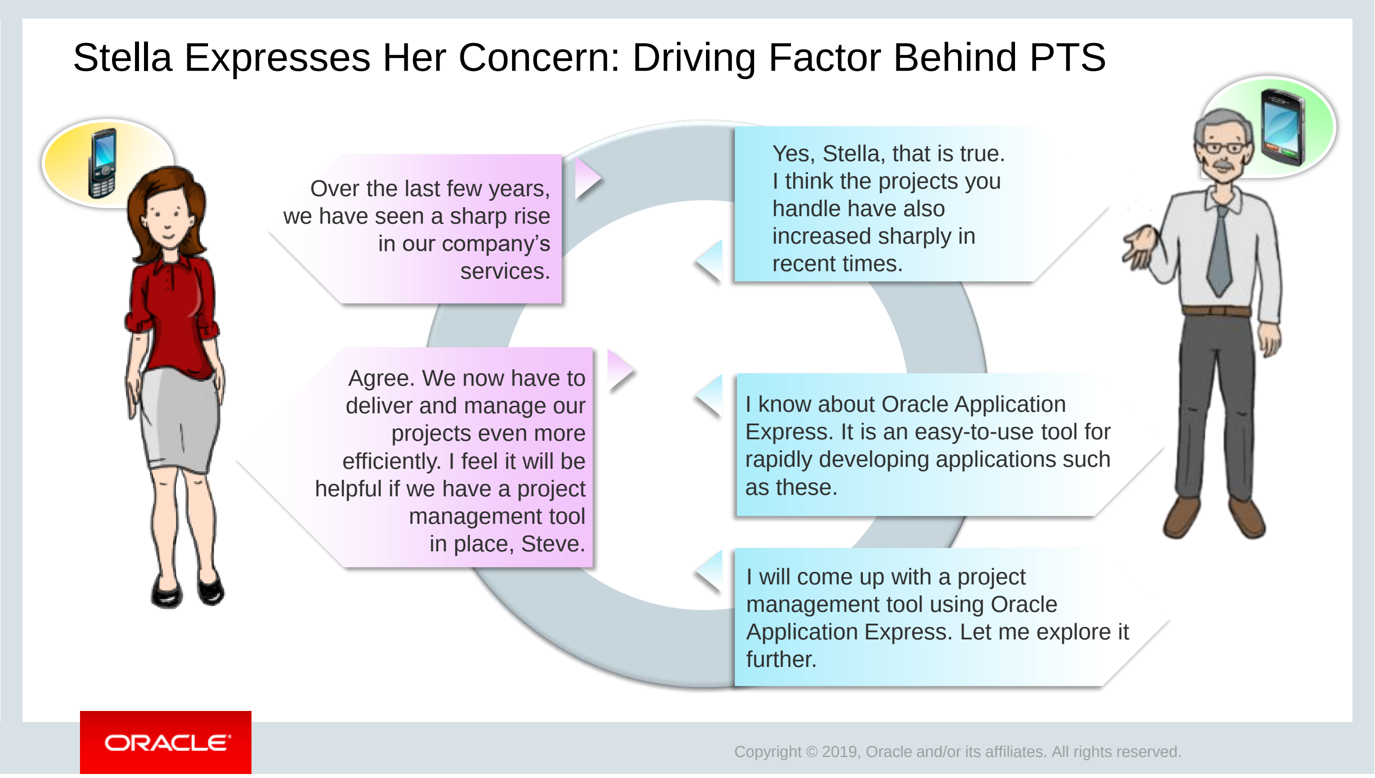
Summary

In this lesson, you should have learned about:

- Course Goals
- Course Road Map
- Course Environment
- Course Persona: Stella and Steve
- Project Tracking System (PTS): Demo Application
- GlobalMart Management Tool (GMT): Practice Application

ORACLE

Unit I Introduction: Getting Started with Application Express



Stella Expresses Her Concern: Driving Factor Behind PTS

Over the last few years, we have seen a sharp rise in our company's services.

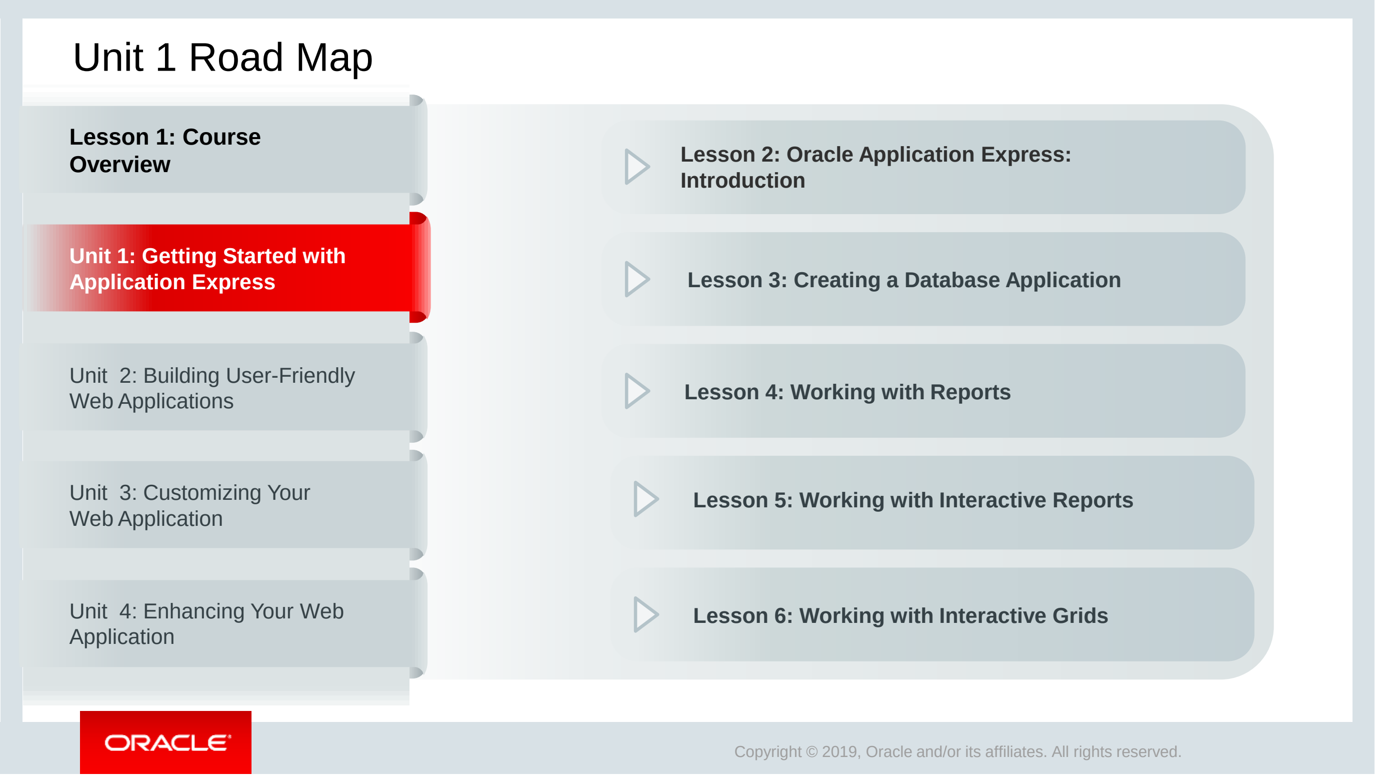
Yes, Stella, that is true. I think the projects you handle have also increased sharply in recent times.

Agree. We now have to deliver and manage our projects even more efficiently. I feel it will be helpful if we have a project management tool in place, Steve.

I know about Oracle Application Express. It is an easy-to-use tool for rapidly developing applications such as these.

I will come up with a project management tool using Oracle Application Express. Let me explore it further.

Stella expresses her concern about the need for a project management tool in their company. Steve thinks of leveraging Oracle's Application Express tool to build a Project Tracking System (PTS) in a short span of time. He decides to explore the basic features and components of Oracle Application Express.



Unit 1 Road Map

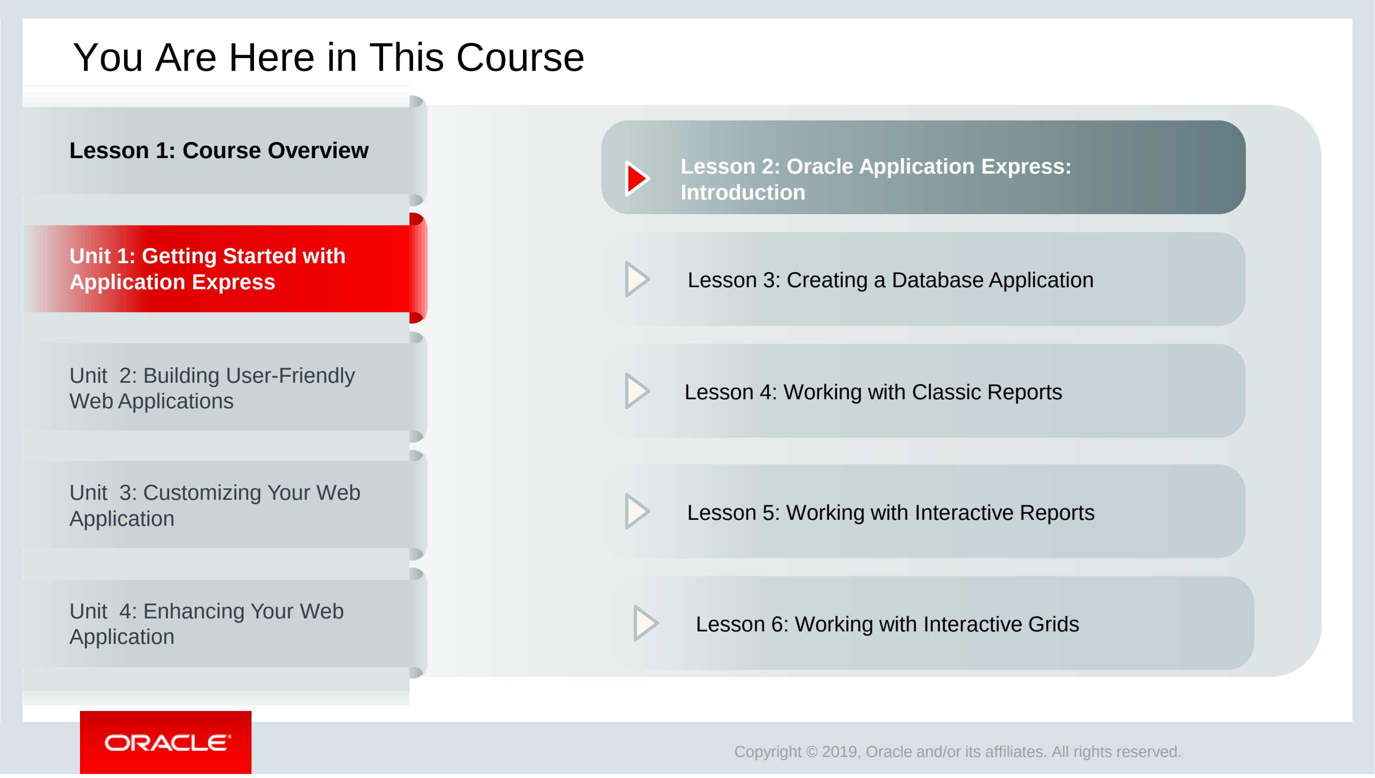
- Lesson 1: Course Overview
- Unit 1: Getting Started with Application Express
  - Lesson 2: Oracle Application Express: Introduction
  - Lesson 3: Creating a Database Application
  - Lesson 4: Working with Reports
  - Lesson 5: Working with Interactive Reports
  - Lesson 6: Working with Interactive Grids
- Unit 2: Building User-Friendly Web Applications
- Unit 3: Customizing Your Web Application
- Unit 4: Enhancing Your Web Application

This slide is a graphical representation of the overall course structure, and Unit 1 in particular.

In Unit 1, you learn how to create database applications using Oracle Application Express 19.1 and create reports and an interactive grid for the database application.

2

Oracle Application Express: Introduction



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

Lesson 2: Oracle Application Express: Introduction

Lesson 3: Creating a Database Application

Lesson 4: Working with Classic Reports

Lesson 5: Working with Interactive Reports

Lesson 6: Working with Interactive Grids

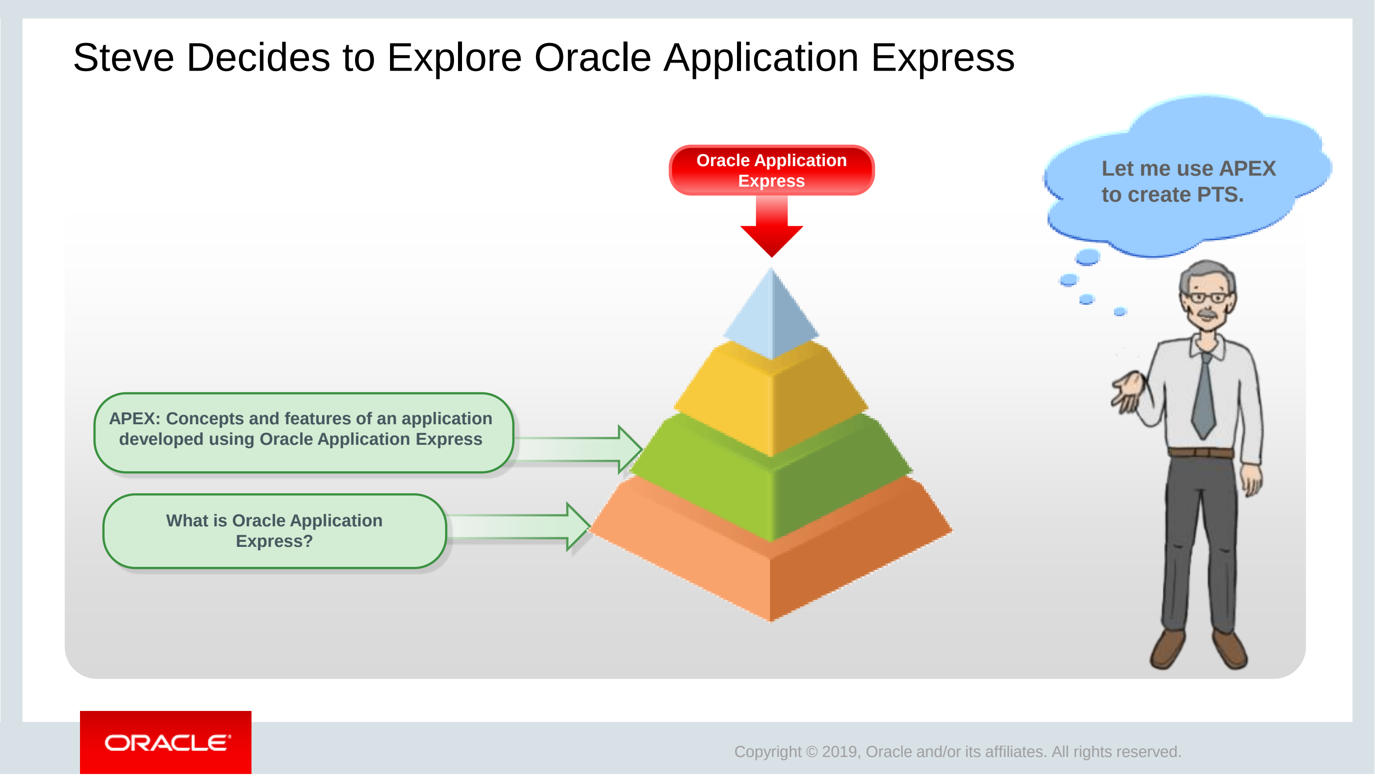
This slide is a graphical depiction of the course, highlighting Unit 1 - Lesson 2.

Objectives

After completing this lesson, you should be able to:

- Describe Oracle Application Express
- Explain Oracle Application Express concepts
- Identify the components of Oracle Application Express
- Run a sample application
- Install a packaged application
- Export and import applications
- Install sample datasets

This lesson introduces you to Oracle Application Express. You identify the key features, benefits, and components of Oracle Application Express. You understand how Oracle Application Express works by learning about its architecture. You get started with Oracle Application Express by setting up the users and the environment used in this course. You will learn how to run a sample application, install and uninstall a packaged application, and install the sample datasets that are provided by default.



Steve Decides to Explore Oracle Application Express

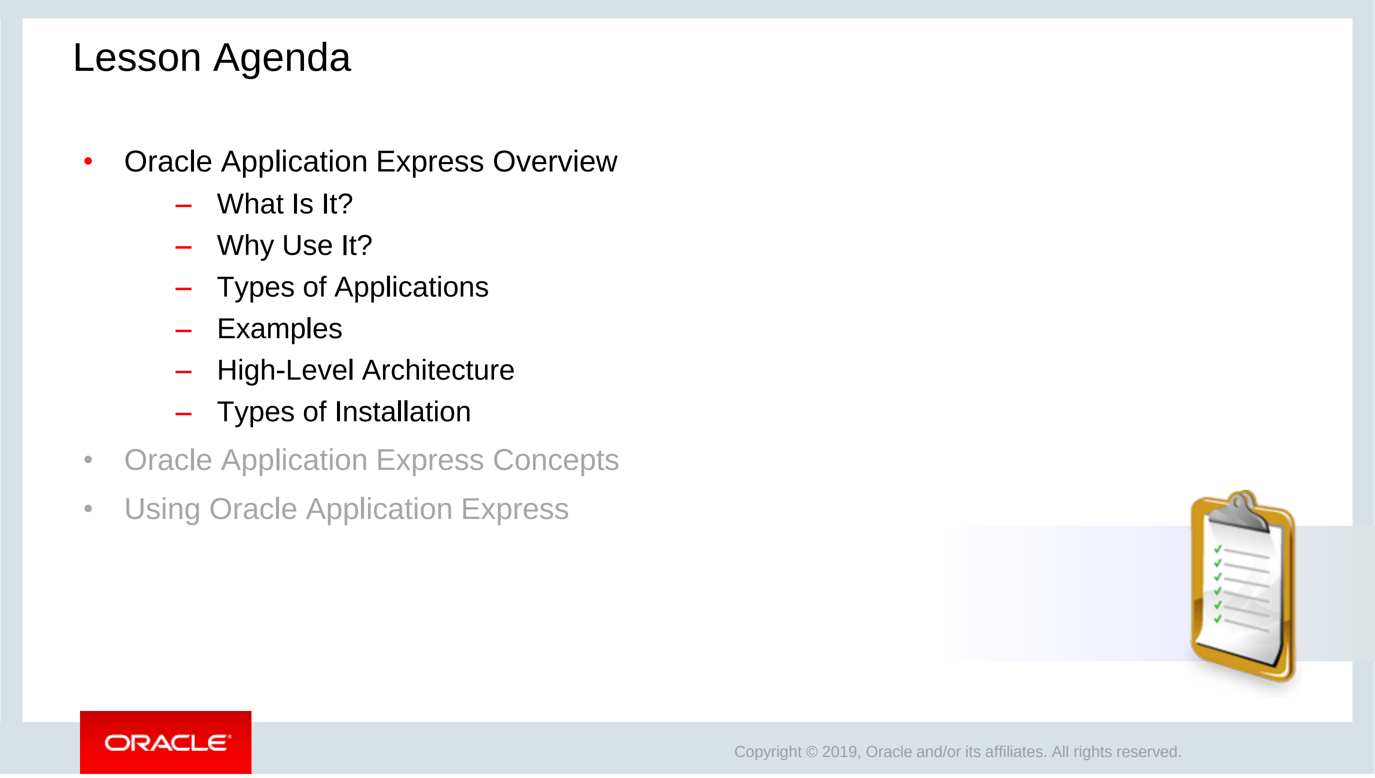
Oracle Application Express

Let me use APEX to create PTS.

APEX: Concepts and features of an application developed using Oracle Application Express

What is Oracle Application Express?

Steve, who is an Application Developer at Project Tracking Corporation, decides to use Oracle Application Express to develop a new application that will address the project management requirements proposed by Stella. He calls this application Project Tracking System (PTS). Steve has been using Oracle Application Express for quite some time, and now he decides to explore it further and use it to develop the application.



Lesson Agenda

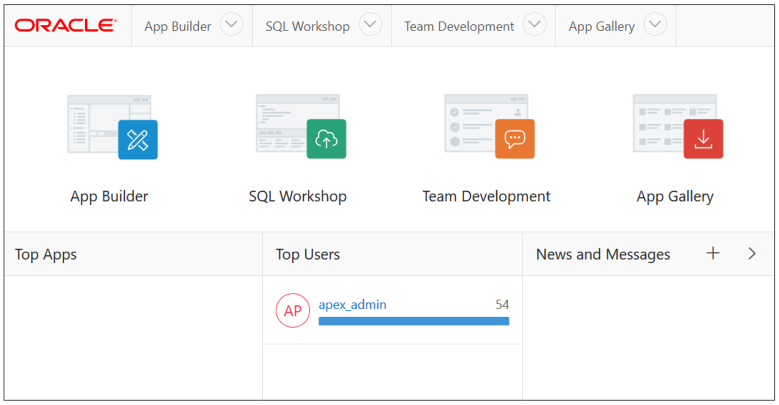
- Oracle Application Express Overview
  - What Is It?
  - Why Use It?
  - Types of Applications
  - Examples
  - High-Level Architecture
  - Types of Installation
- Oracle Application Express Concepts
- Using Oracle Application Express

ORACLE

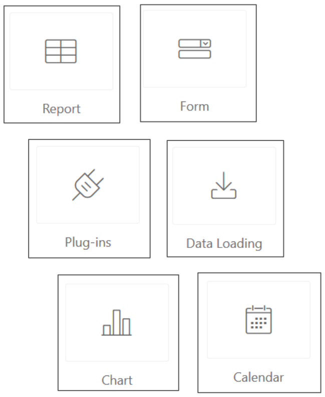
What Is Oracle Application Express?

Oracle Application Express is a web application development, deployment, and maintenance tool.

Oracle Application Express Home Page



Key Features



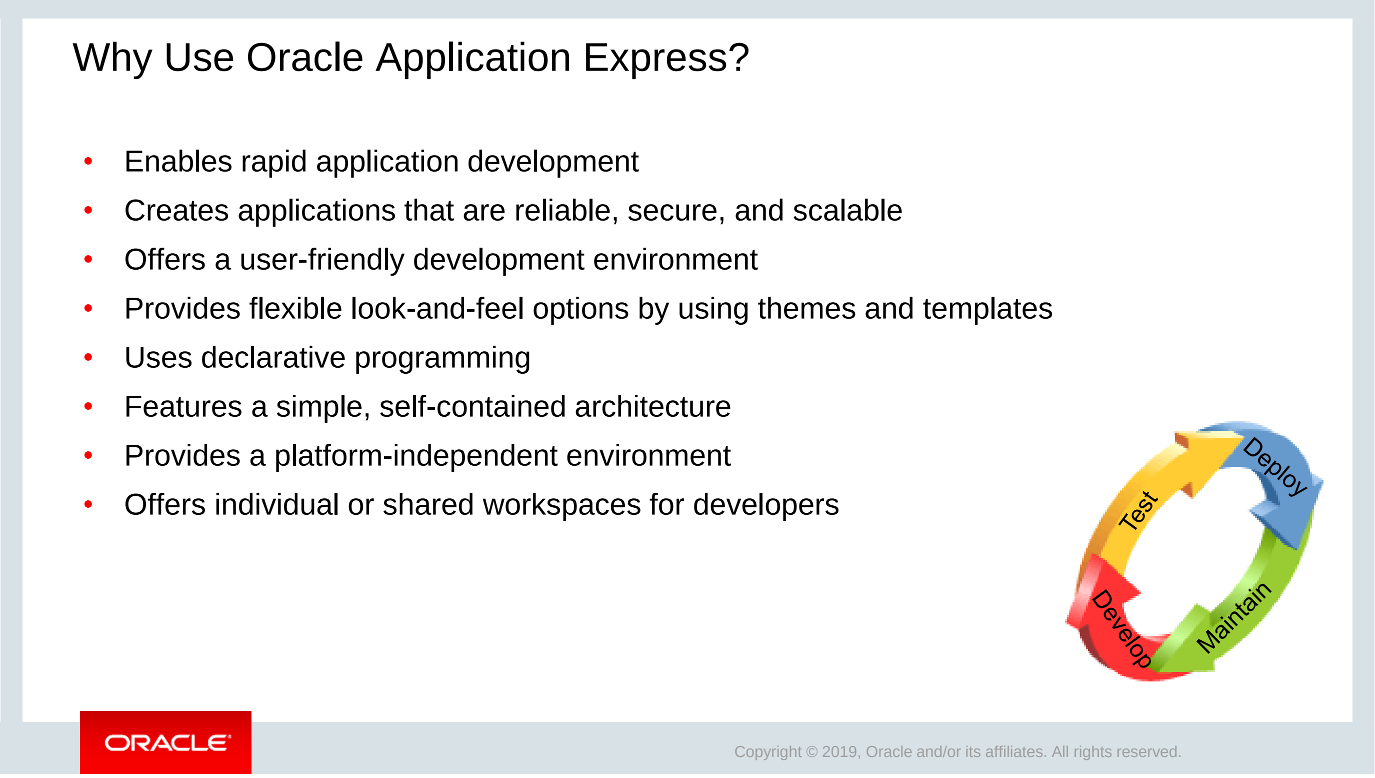
Stella has cited the need for a purely project management tool that will help her track multiple projects, bugs, and employees, manage feedback, and so on. In this scenario, Oracle Application Express is the best fit to meet Stella’s requirements because it provides all the functionalities to create a product to track and manage projects.

Oracle Application Express is a web-based development and deployment tool that is available with Oracle Database. You can create database-centric web applications that are reliable, scalable, and secure. The following are some key highlights of Oracle Application Express:

- It has several built-in features and wizards that quicken your application development process. Some of the key features are listed in this slide.
- It has a user-friendly graphical interface, which makes it very intuitive and easy to use.
- It requires minimal programming knowledge to use Oracle Application Express.

The application definition is stored as metadata in the Oracle Database tables. When you run your application, the Oracle Application Express engine assembles the pages from the database and displays them in your browser.

Oracle Application Express was first released in 2004, and it was called HTML DB then.

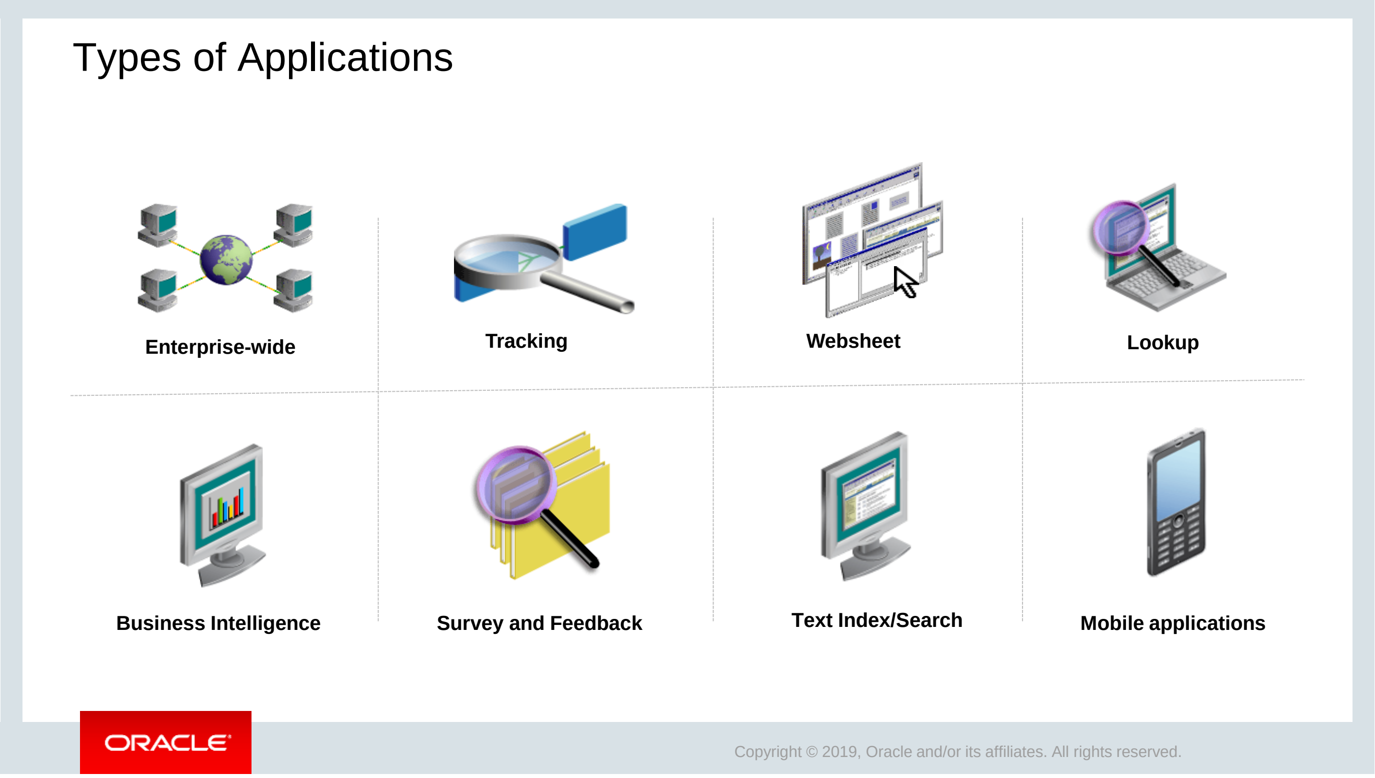


Why Use Oracle Application Express?

- Enables rapid application development
- Creates applications that are reliable, secure, and scalable
- Offers a user-friendly development environment
- Provides flexible look-and-feel options by using themes and templates
- Uses declarative programming
- Features a simple, self-contained architecture
- Provides a platform-independent environment
- Offers individual or shared workspaces for developers

Oracle Application Express comes with a number of advantages that clearly indicate its edge over other development and deployment tools:

- By using Oracle Application Express, you can develop web-based, database-centric applications that are reliable and fast, as well as secure and scalable. It has a user-friendly interface, enabling you to create and deploy applications in a short span of time. You can use the available themes and templates to customize your application.
- Oracle Application Express uses a declarative framework for web application development. This means that you specify what to do rather than how to do it. No code is generated or compiled. You interact with wizards and property sheets to define your application.
- It enables organizations to capitalize on their existing investment in SQL and PL/SQL skills. Few programming skills are required, and anyone can quickly learn to develop applications. With Oracle Application Express, you can build applications with minimal programming knowledge, with fewer developers.
- It can be installed on a single workstation or on a server that can support multiple developers. An administrator centrally manages and administers the development environment and creates a shared workspace in a single installation. The definition of an entire application can be easily packaged and exported for deployment and installation into another Oracle Application Express instance.



Types of Applications

Enterprise-wide

Tracking

Websheet

Lookup

Business Intelligence

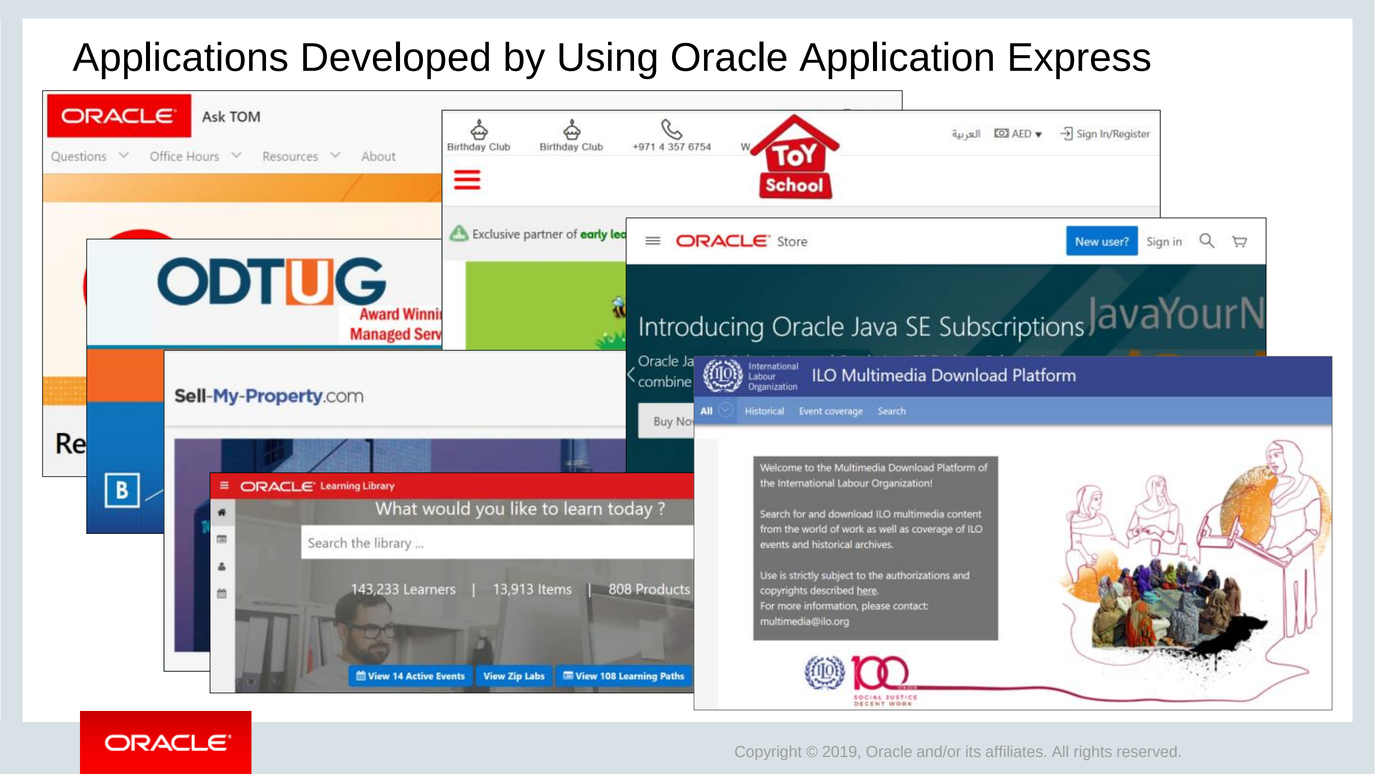
Survey and Feedback

Text Index/Search

Mobile applications

The following are examples of the types of applications that are developed by using Oracle Application Express:

- Enterprise-wide applications
- Web-based applications to track projects, contacts, customers, leads, and assets
- Websheet applications that enable end users to manage structured and unstructured data without developer assistance
- Applications to look up people and catalog items
- Lightweight Business Intelligence (BI) applications with reports, bar charts, line charts, and pie charts. These applications can be based on summarized data copied from a live database or operate on live transaction data. The charts and reports enable drilling down and cross-referencing of information.
- Web-based applications that use the text indexing and search capability of Oracle Database
- Applications that must be built in a very short span of time (usually a week)
- Mobile applications

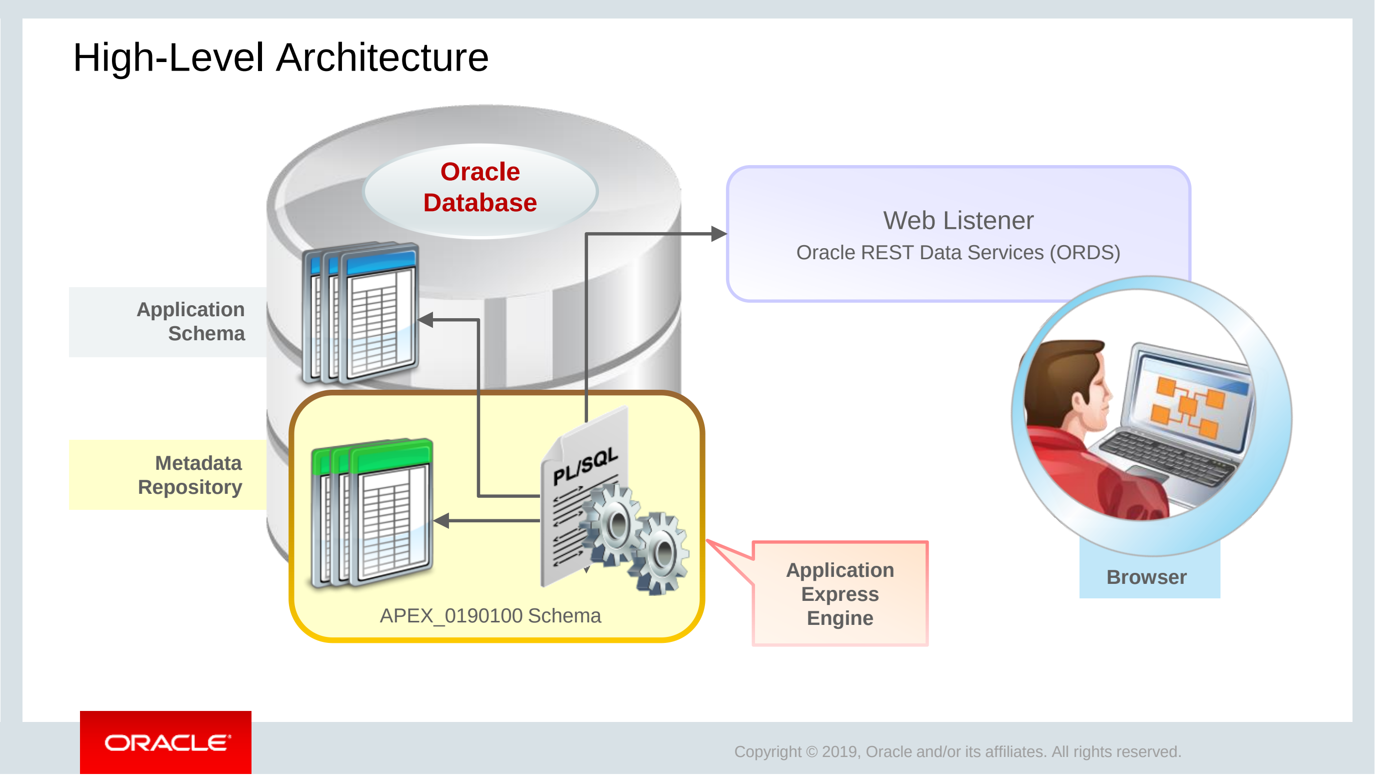


Applications Developed by Using Oracle Application Express

ORACLE Ask TOM
Questions
Office Hours
Resources
About
ODTUG
Award Winni
Managed Serv
Re
B
Birthday Club
Birthday Club
+971 4 357 6754
ToY School
العربية
AED
Sign In/Register
Exclusive partner of early lea
ORACLE Store
New user?
Sign in
Introducing Oracle Java SE Subscriptions
JavaYourN
Oracle Ja
combine
Buy Now
Sell-My-Property.com
ORACLE Learning Library
What would you like to learn today ?
Search the library ...
143,233 Learners | 13,913 Items | 808 Products
View 14 Active Events
View Zip Labs
View 108 Learning Paths
International Labour Organization
ILO Multimedia Download Platform
All
Historical
Event coverage
Search
Welcome to the Multimedia Download Platform of the International Labour Organization!
Search for and download ILO multimedia content from the world of work as well as coverage of ILO events and historical archives.
Use is strictly subject to the authorizations and copyrights described here.
For more information, please contact: multimedia@ilo.org
100
SOCIAL JUSTICE DECENT WORK
ORACLE

The slide shows a variety of applications that have been developed by using Oracle Application Express.

Note: The user interface of the application is also developed using Oracle Application Express.



High-Level Architecture

Now that you have an idea about the versatility of Oracle Application Express, and the types of applications that you can create using Oracle Application Express, let's look at its architecture.

High-Level Architecture

When you install Oracle Application Express 19.1, it resides in your Oracle database as a separate schema named `APEX_0190100.` This schema houses Oracle Application Express that consists of:

- Metadata stored in database tables
- The Application Express engine, which is written by using PL/SQL code

When you create an application, its definition is stored in the metadata repository. At every stage of application development, metadata is created or modified and stored in the repository tables. The Application Express engine assembles the application pages by accessing the metadata repository.

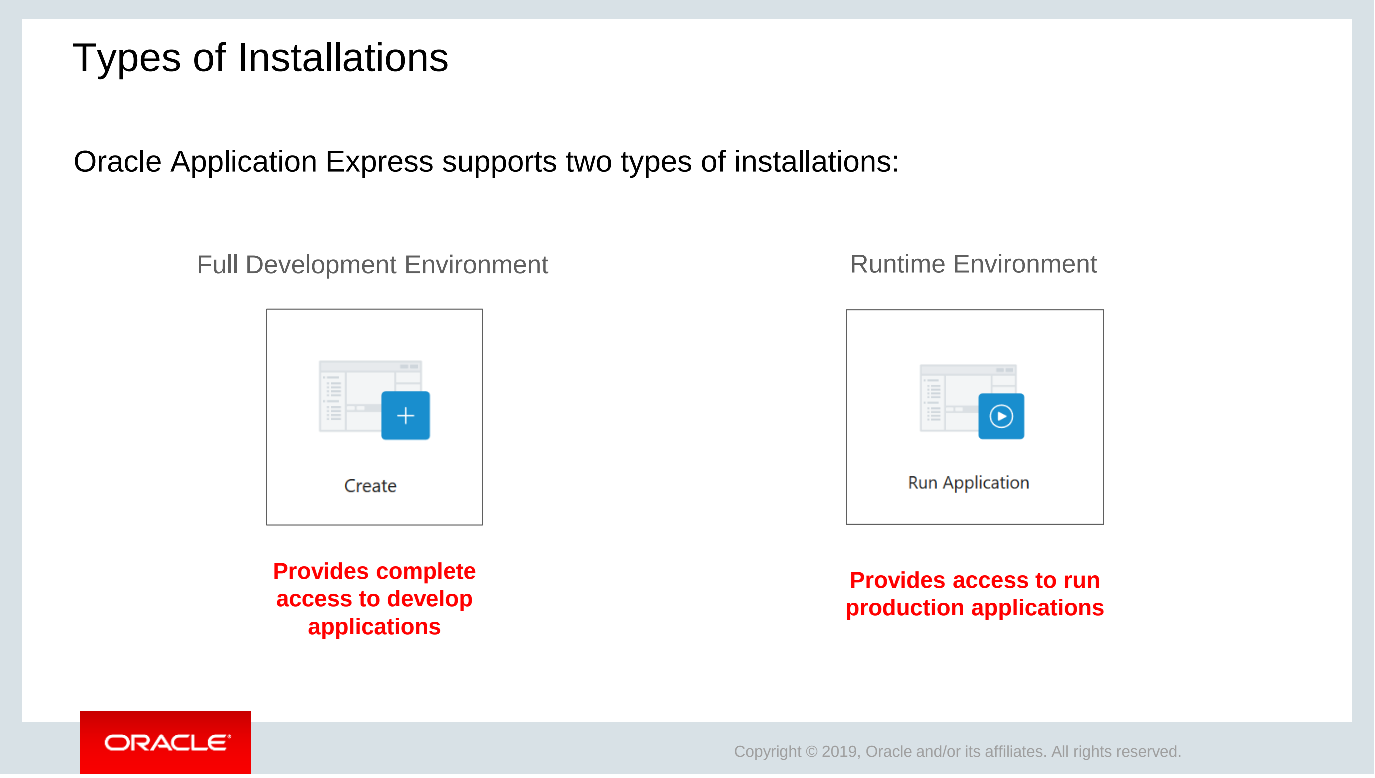
When you run your application from the browser, calls are made to the Application Express engine. The engine then processes and renders the application components in real time, based on the data in the metadata repository and the schema against which the application is running.

To enable your web browser to interact with the Application Express engine, you need a web listener, which acts as a communication facilitator between the web browser and Oracle Application Express. It maps browser requests to database stored procedure calls.

With Oracle Application Express 19.1, you have this choice for web listener:

- **Oracle REST Data Services (formerly known as APEX Listener):** Also known as ORDS, it's a Java-based web listener, which is compatible with any J2EE-compliant web server. This is the most recommended web listener option from Oracle. It not only acts as a web listener but also helps in integrating your applications with Web Services.

The practice environment for this course uses the Oracle REST Data Services (ORDS) as the web listener.



Types of Installations

Oracle Application Express supports two types of installations:

Full Development Environment

Create

Provides complete access to develop applications

Runtime Environment

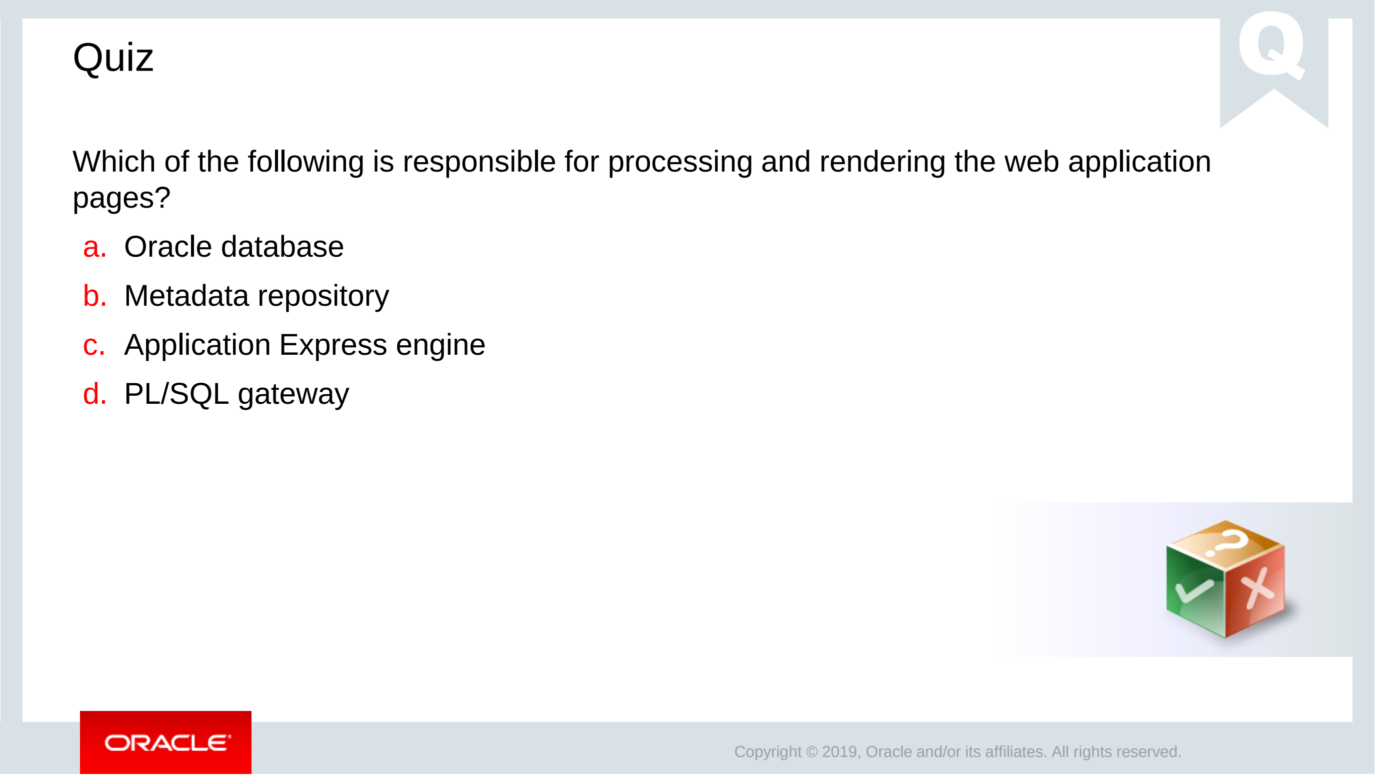
Run Application

Provides access to run production applications

Based on your requirements, you can install Oracle Application Express in one of the following ways:

- **Full development environment:** This installation provides complete access to the Application Builder environment to develop applications.
- **Runtime environment:** This installation is an appropriate choice for production implementations in which you want to run applications that cannot be modified.

An Oracle Application Express runtime environment enables you to run production applications. But it does not provide a web interface for administration. The runtime environment installation option minimizes the installed footprint and privileges. In a runtime instance, developers cannot inadvertently update a production application. Therefore, the runtime environment improves application security.

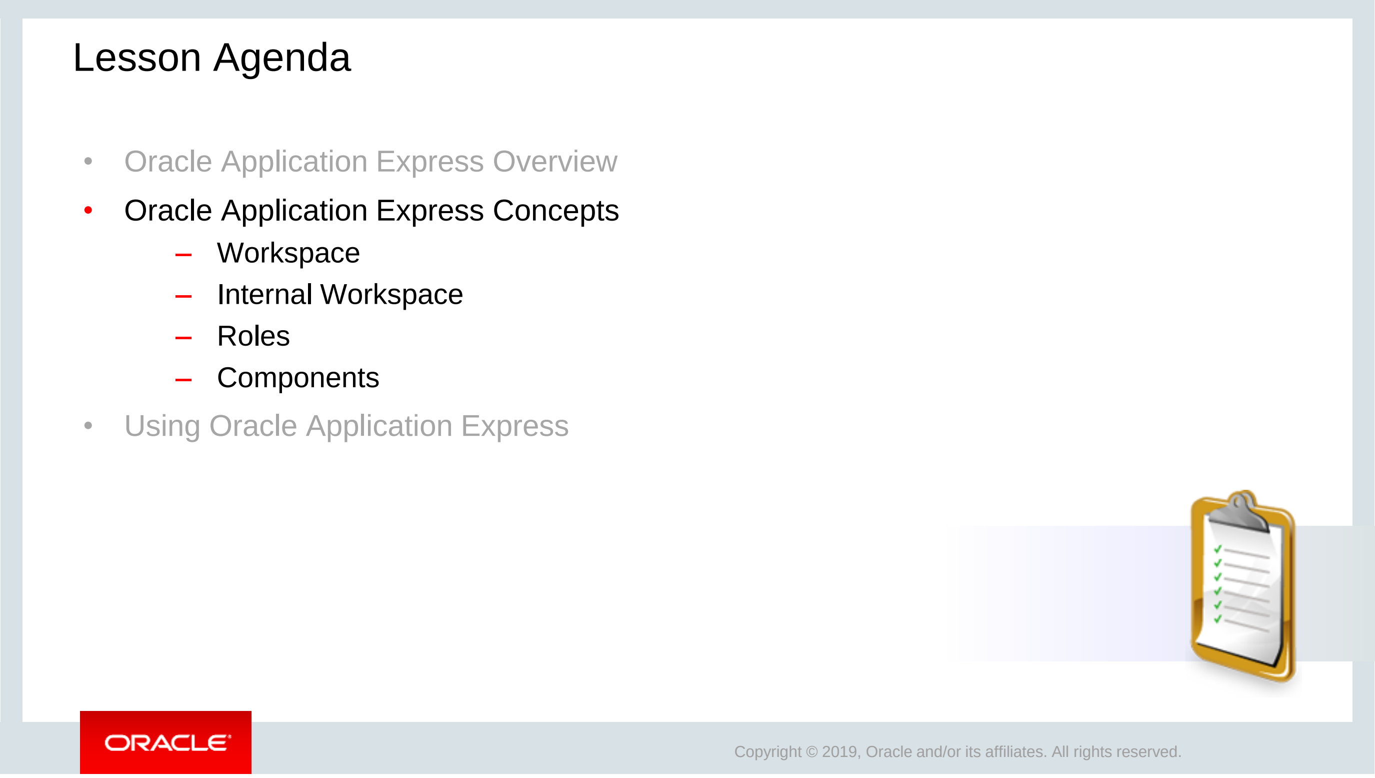


Quiz

Which of the following is responsible for processing and rendering the web application pages?

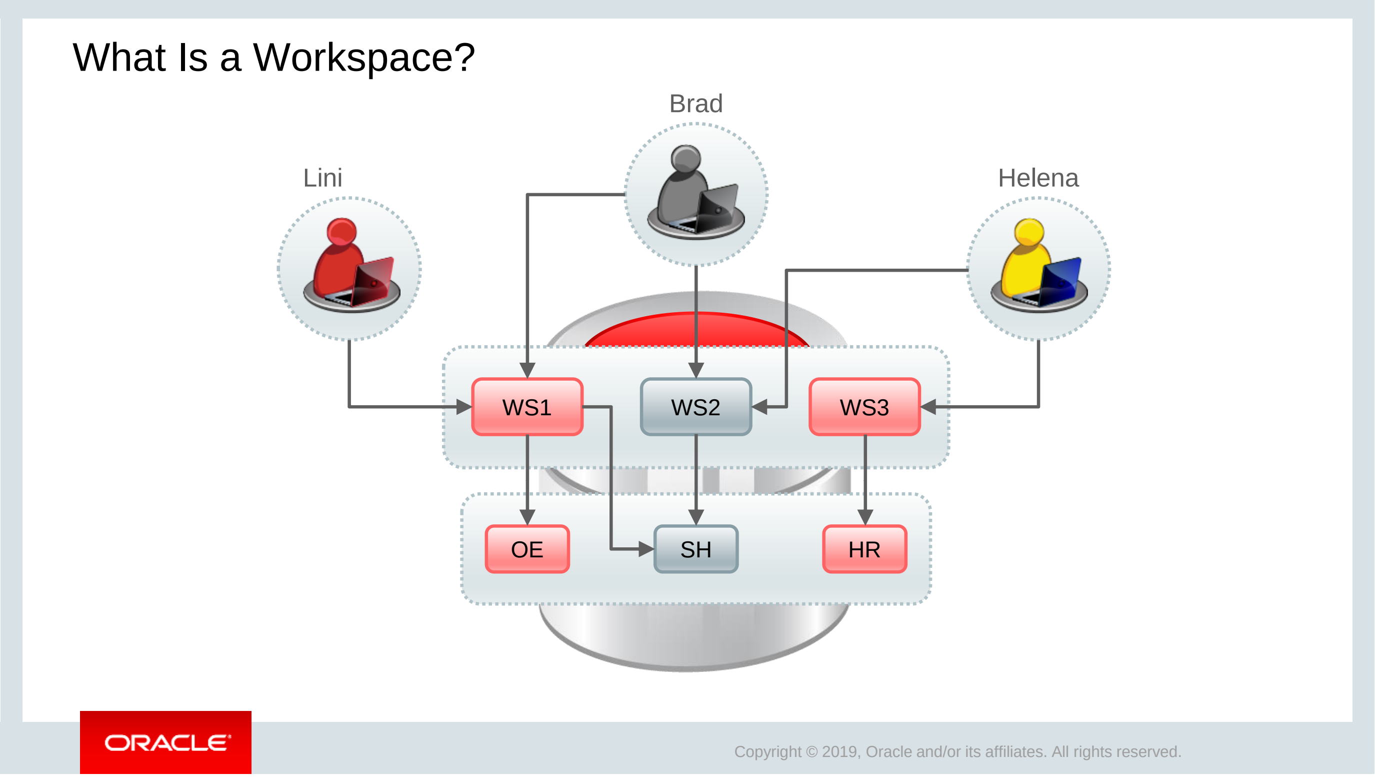
a. Oracle database
b. Metadata repository
c. Application Express engine
d. PL/SQL gateway

Answer: c



Lesson Agenda

- Oracle Application Express Overview
- Oracle Application Express Concepts
  - Workspace
  - Internal Workspace
  - Roles
  - Components
- Using Oracle Application Express



What Is a Workspace?

To sign in to Oracle Application Express, you need a workspace and the credentials for that workspace. A workspace is a logical application development environment, where you define applications. Within a single Oracle Application Express installation, you can define any number of workspaces, for different departments, different projects, and so forth. Each workspace can be associated with one or more data schemas, where the database objects and data are stored, and on which the applications are generally built.

To create an application, you must first create or have access to a workspace. By associating a workspace with a schema, you can:

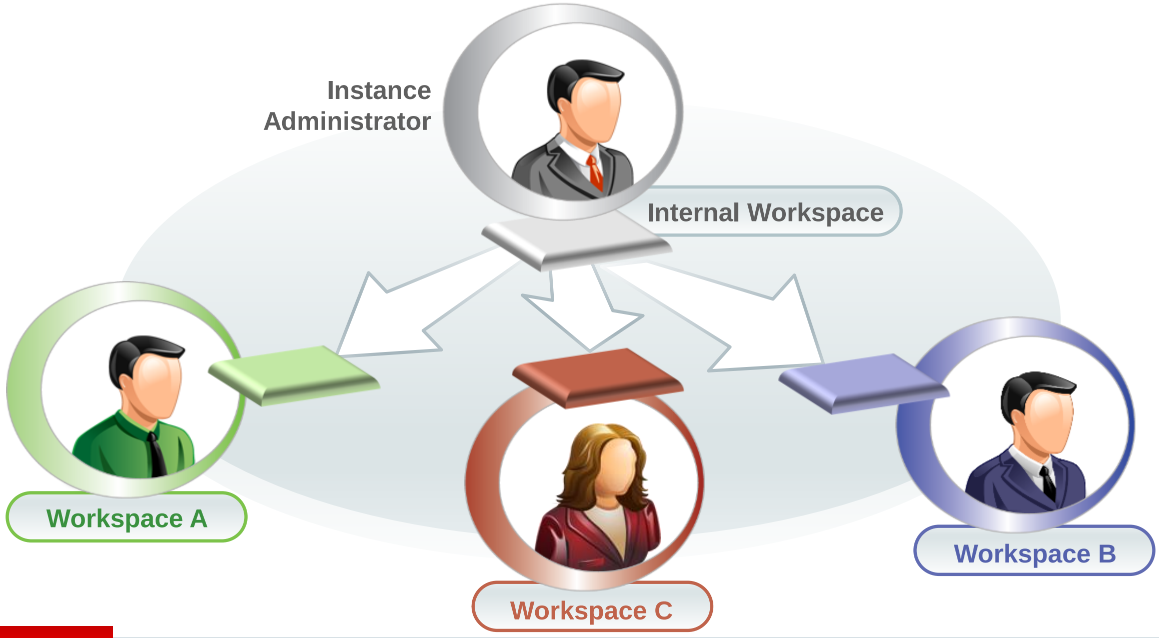
- Build applications that interact with the database objects in that schema
- Create new database objects in that schema

A single workspace supports working with more than one schema. One or more developers or end users can access a workspace.

As shown in the graphic in the slide, a single Oracle Database instance can contain multiple Oracle Application Express workspaces. In this example, you see Lini, Brad, and Helena, the three developers working with Steve, along with three different workspaces (WS1, WS2, and WS3). Lini and Brad have access to WS1. Besides, Brad also has access to WS2. Helena has access to WS2 and WS3. Each workspace has access to one or more database schemas. For example, WS1 has access to `OE` and `SH` schemas, WS2 has access to `SH`, and WS3 has access to `HR`. Multiple developers can work by using the same database instance from different workspaces or the same workspace with access to the same or different schema.

Thus, Oracle Application Express turns a single Oracle Database instance into a shared workgroup database service. This service can be accessed through a browser with no installation required on the desktop for both the developer and the end user.

What Is an Internal Workspace?



An internal workspace is a special workspace, a virtual area, that is created by default when Oracle Application Express is installed. An internal workspace is:

- Accessible only to instance administrators
- Used to create and manage workspaces in the Oracle Application Express instance

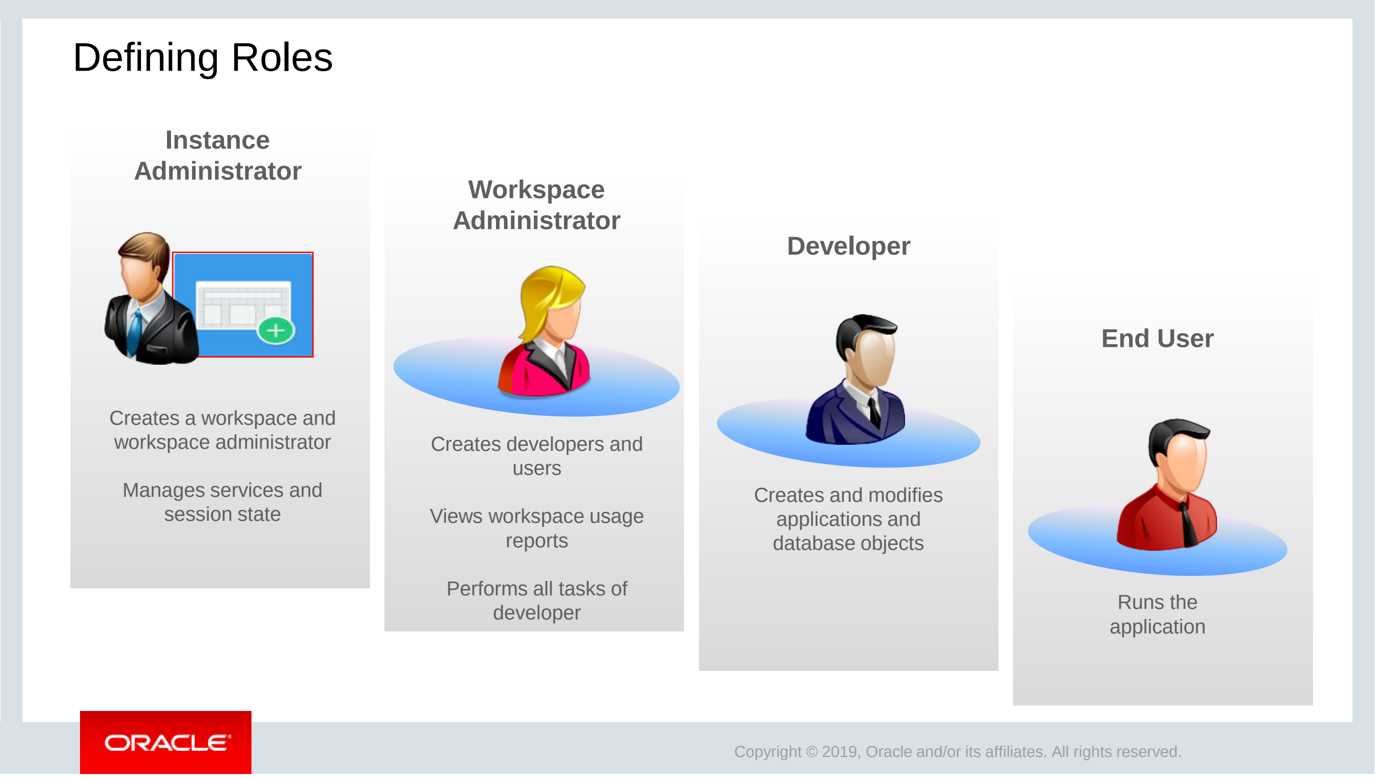
To sign in to the internal workspace, enter the following URL in the address bar:

`http://<hostname>:<port>/apex/apex_admin.`

On the **Sign In** page that opens, enter the following credentials:

- Username: `admin`
- Password: Enter the password that was set up during installation.

You can learn more about how to perform administration tasks by using the internal workspace in the *Oracle Application Express: Administration* course.



Defining Roles

Instance Administrator

Creates a workspace and workspace administrator

Manages services and session state

Workspace Administrator

Creates developers and users

Views workspace usage reports

Performs all tasks of developer

Developer

Creates and modifies applications and database objects

End User

Runs the application

In the Oracle Application Express workspace, different users with different roles perform various tasks. Let us understand these different roles and their privileges that are defined in Oracle Application Express. Four roles are defined in Oracle Application Express:

- Instance administrator
- Workspace administrator
- Developer
- End user

Instance Administrator

An *Instance Administrator* manages the entire Oracle Application Express instance, including service administration and workspace administration. The *Instance Administrator* manages the workspaces of all users and is also responsible for monitoring usage and managing session state, a functionality that enables developers to store and retrieve values for a user as the user navigates between different application pages. The default Oracle Application Express administration privileged user is `admin`.

The *Instance Administrator* performs the following tasks:

1. Signs in to Oracle Application Express as an administrator
2. Creates a workspace and a workspace administrator. Both can be done at the same time by using the Create Workspace Wizard.

Workspace Administrator

A *Workspace Administrator* has administrative privileges for a workspace and can add and delete new users to the workspace, create new user groups, and view usage reports of the workspace.

The *Workspace Administrator* performs the following tasks:

1. Signs in to Oracle Application Express by using the workspace that has been assigned by the Instance Administrator
2. Creates *developer* users for the workspace
3. Installs sample applications
4. Installs a packaged application with supporting objects

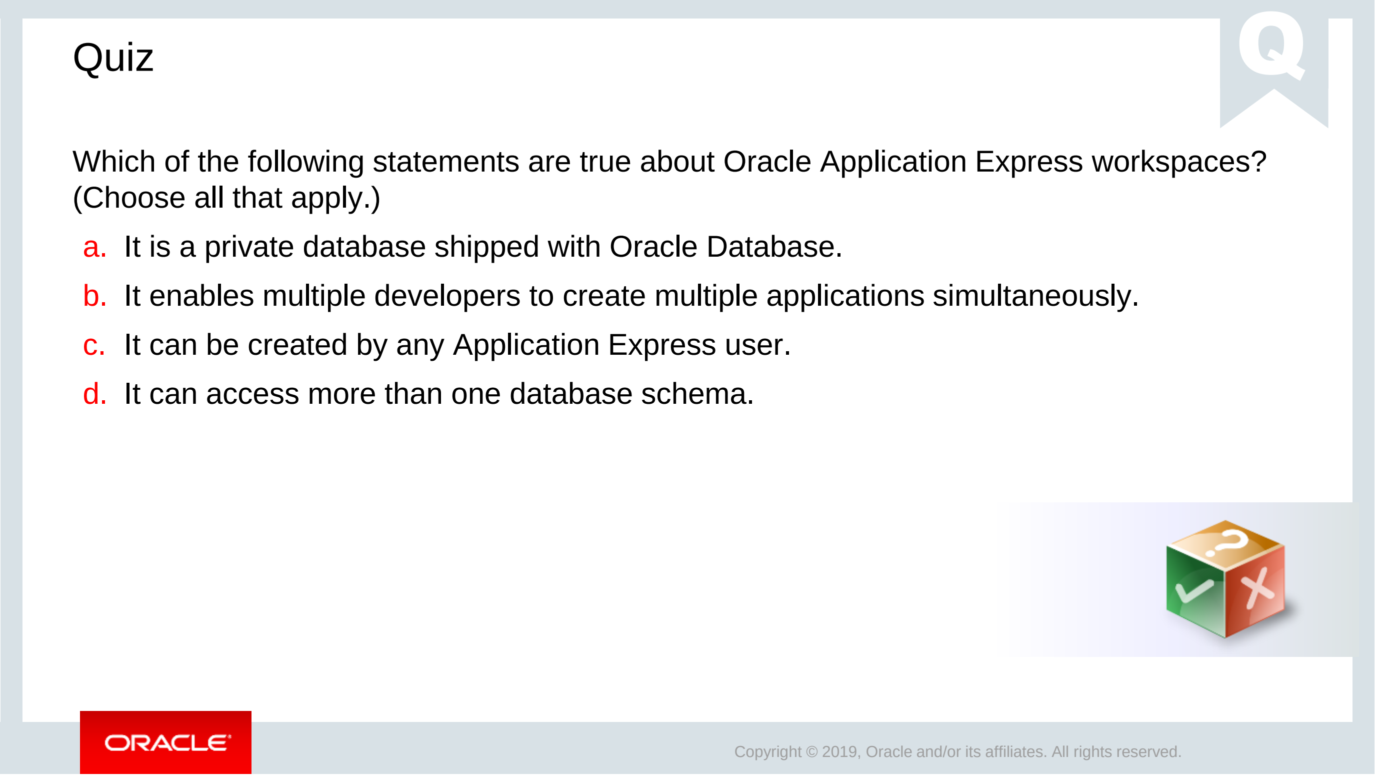
Developer

A *developer* has development privilege and can sign in to a workspace through which he or she can access his or her own database objects. In addition to having private workspaces, a *developer* can also share a workspace to develop applications.

Multiple *developers* can sign in to the same Oracle Application Express instance to develop and edit applications.

End User

The *end user* is a user without development and administration privileges. This user has only the basic privileges needed to run an application.



Quiz

Which of the following statements are true about Oracle Application Express workspaces? (Choose all that apply.)

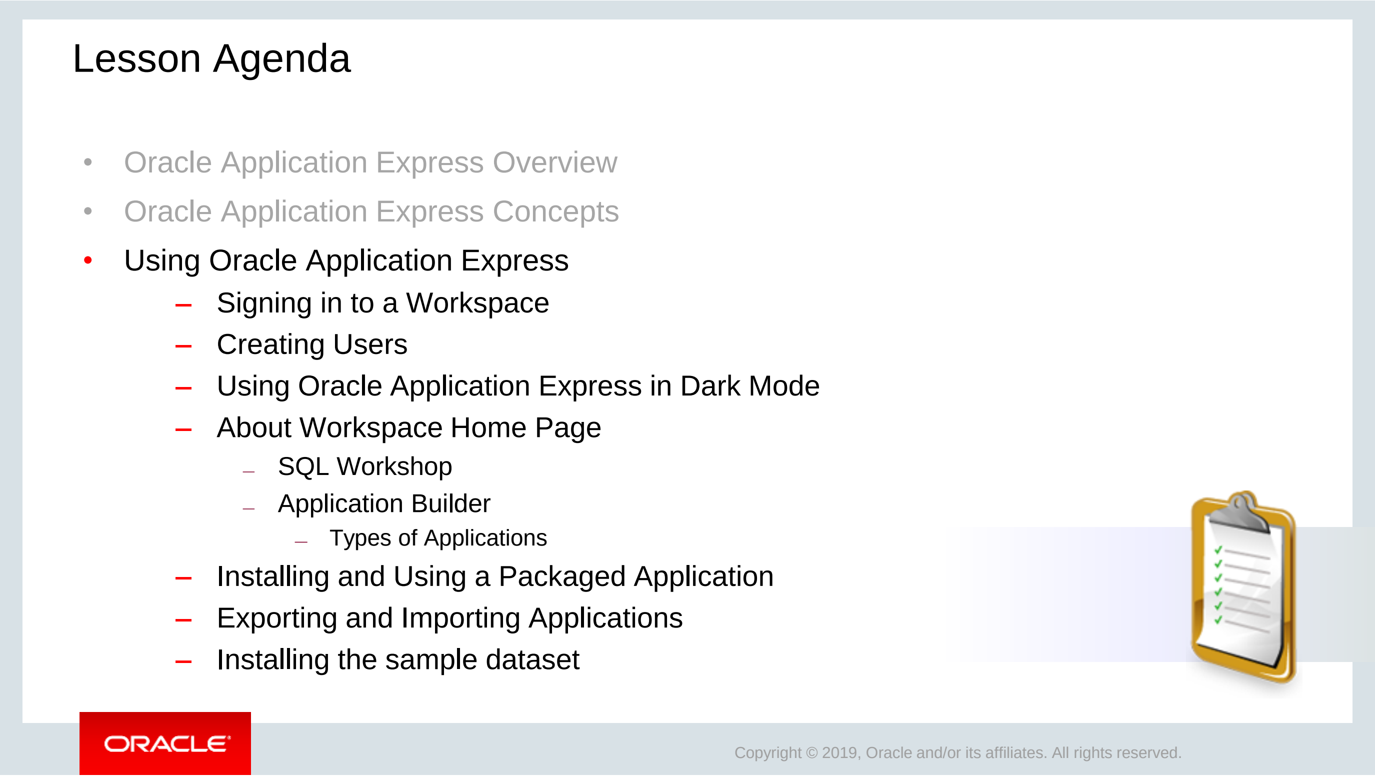
a. It is a private database shipped with Oracle Database.

b. It enables multiple developers to create multiple applications simultaneously.

c. It can be created by any Application Express user.

d. It can access more than one database schema.

Answer: b, d



Lesson Agenda

- Oracle Application Express Overview
- Oracle Application Express Concepts
- Using Oracle Application Express
  - Signing in to a Workspace
  - Creating Users
  - Using Oracle Application Express in Dark Mode
  - About Workspace Home Page
    - SQL Workshop
    - Application Builder
      - Types of Applications
  - Installing and Using a Packaged Application
  - Exporting and Importing Applications
  - Installing the sample dataset

ORACLE

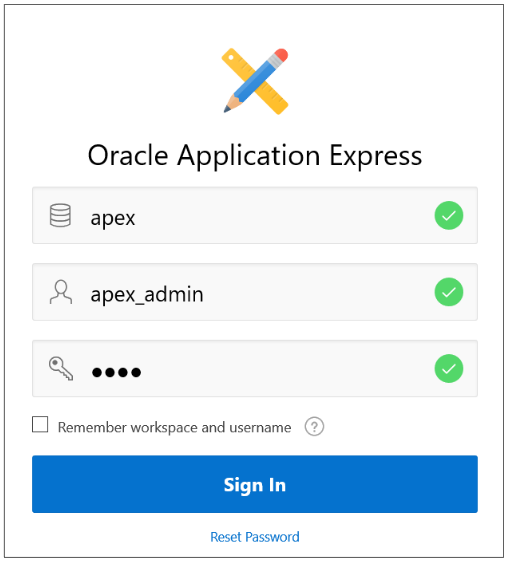
You will now sign in to Oracle Application Express and perform the following tasks. Performing these tasks will provide you a hands-on experience with Oracle Application Express.

- Create a developer user.
- Run the sample database application.
- Install a packaged app.
- Install the sample dataset.

Signing In to a Workspace

To sign in to an Oracle Application Express workspace:

1. Enter the correct URL in your browser address bar.
2. Enter the workspace name.
3. Enter the username and password. Then click **Sign In.**



Oracle Application Express

apex

apex_admin

●●●●

□ Remember workspace and username

To sign in to Oracle Application Express, you need a workspace name and the credentials for that workspace. A workspace is a virtual private database, which enables you to work with multiple users within the same Oracle Application Express installation while keeping your objects, data, and applications private.

You can sign in to Oracle Application Express as a *workspace administrator* or as a *developer* with the following URL:

```
http://<hostname>:<port>/apex
```

On the Sign In page, enter the following credentials:

- **Workspace name:** `apex`
- **Username:** `apex_admin`
- **Password:** `apex`

Click **Sign In.** You may be prompted to change your workspace password the first time you sign in. This option is set when your username and password are created by the Oracle Application Express administrator. You can set your new password to be the same as your old password.

Notes

- If your setup uses Oracle HTTP Server with `mod_plsql`, then use:

  ```
  http://<hostname>:<port>/pls/apex
  ```

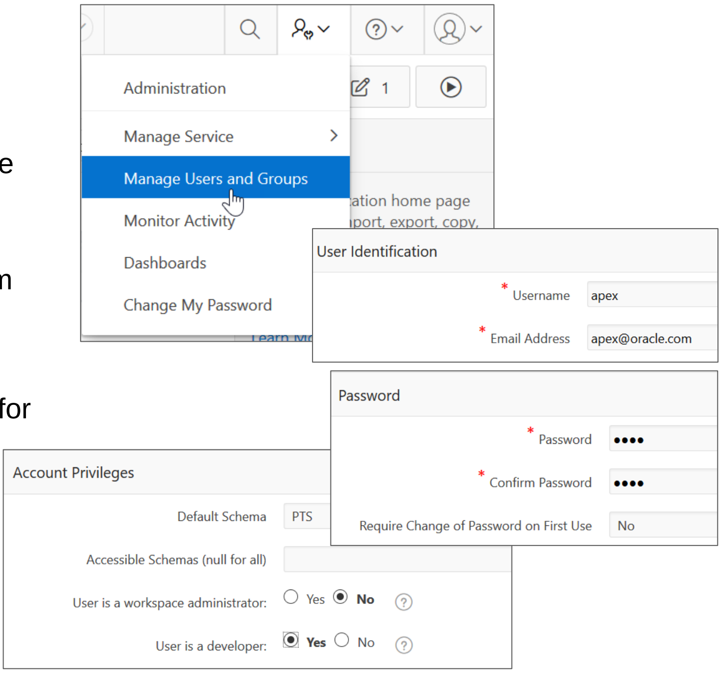
- If your setup uses embedded PL/SQL gateway or APEX Listener, then use:

  ```
  http://<hostname>:<port>/apex
  ```

Creating a Developer User

To create a developer user, perform the following steps:

1. On the Oracle Application Express home page, click the down arrow on the **Administration** tab.
2. Select **Manage Users and Groups** from the drop-down menu.
3. Click the **Create User** button.
4. Enter the username and email address for the user.
5. Review the account privileges for the user.
6. Enter the password for the user.
7. Click the **Create User** button.



User Identification

* Username	apex
* Email Address	apex@oracle.com

Password

* Password	••••
* Confirm Password	••••
Require Change of Password on First Use	No

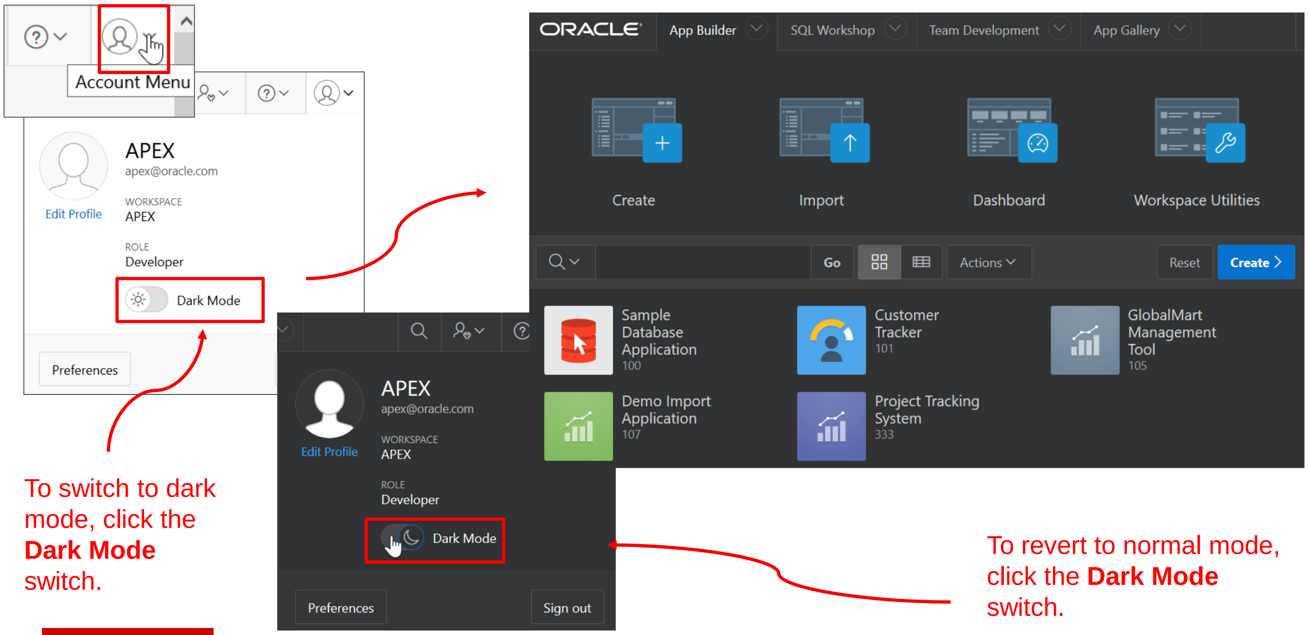
Account Privileges

Default Schema	PTS
Accessible Schemas (null for all)	
User is a workspace administrator:	○ Yes ● No
User is a developer:	● Yes ○ No

This slide provides you an overview of the steps to create a developer user. The Workspace Administrators create the developer users who build applications. To create a developer user:

1. Sign in to Oracle Application Express as a Workspace Administrator.
2. Click the Administration tab and select **Manage Users and Groups** from the drop-down list.
3. On the Manage Users and Groups page, click **Create User.**
4. On the Create User page, enter the following details:
   - **User Identification:** Set the username, name, email ID, and all other details related to the user identification in this section.
   - **Account Privileges:** You can set the default schema for the user. You can restrict access to a specific set of schemas in a workspace or allow access to all schemas. You have an option to give the developer *administrator* privileges. You can also restrict access to the components of Oracle Application Express.
   - **Password:** Set the user password in this section. You can also set the option to change the password on first use.
   - **Group Assignments:** Use this option if you want to use groups for categorization and to manage privileges. You can assign one or all of the following assignments `OAuth2 Client Developer, RESTFul services,` and `SQL Developer` to the user.
5. Click **Create User.**

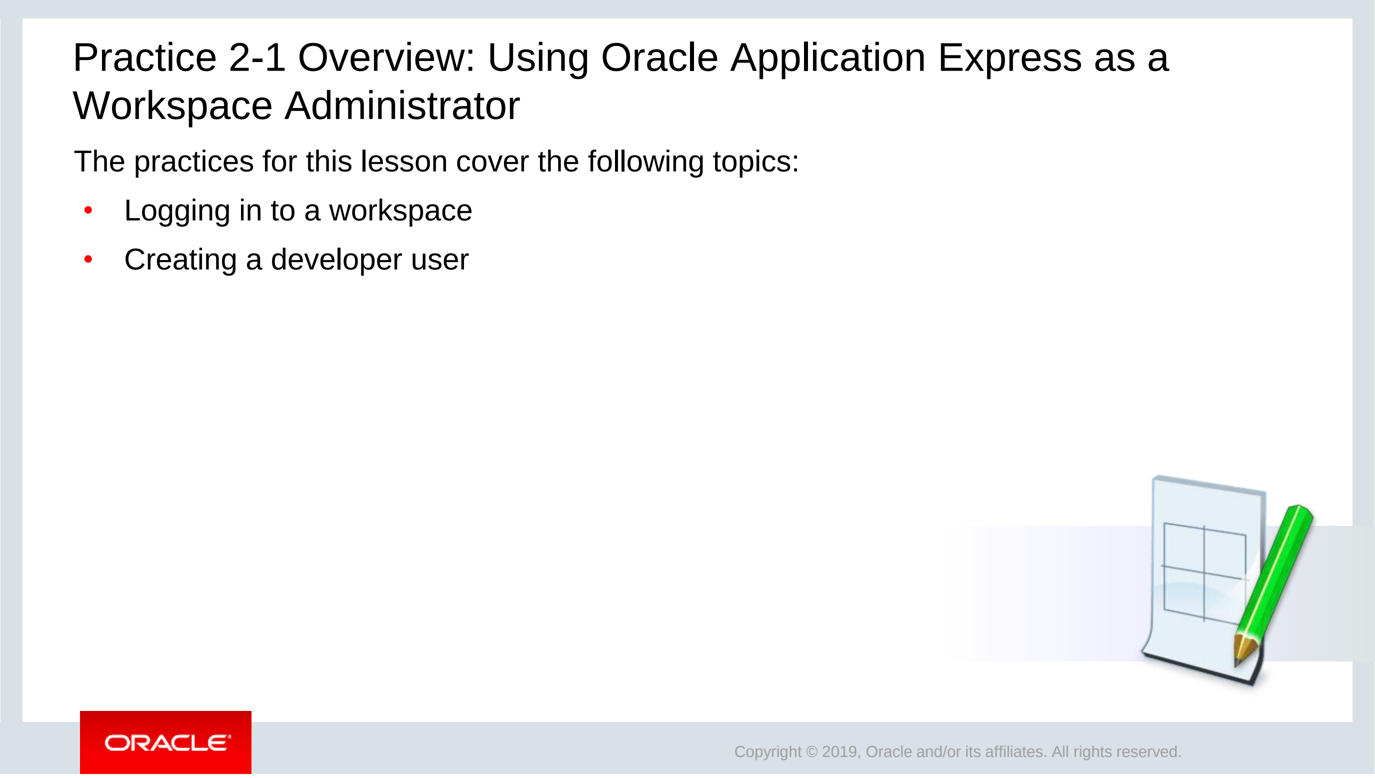
Using Application Express in Dark Mode



To switch to dark mode, click the **Dark Mode** switch.

To revert to normal mode, click the **Dark Mode** switch.

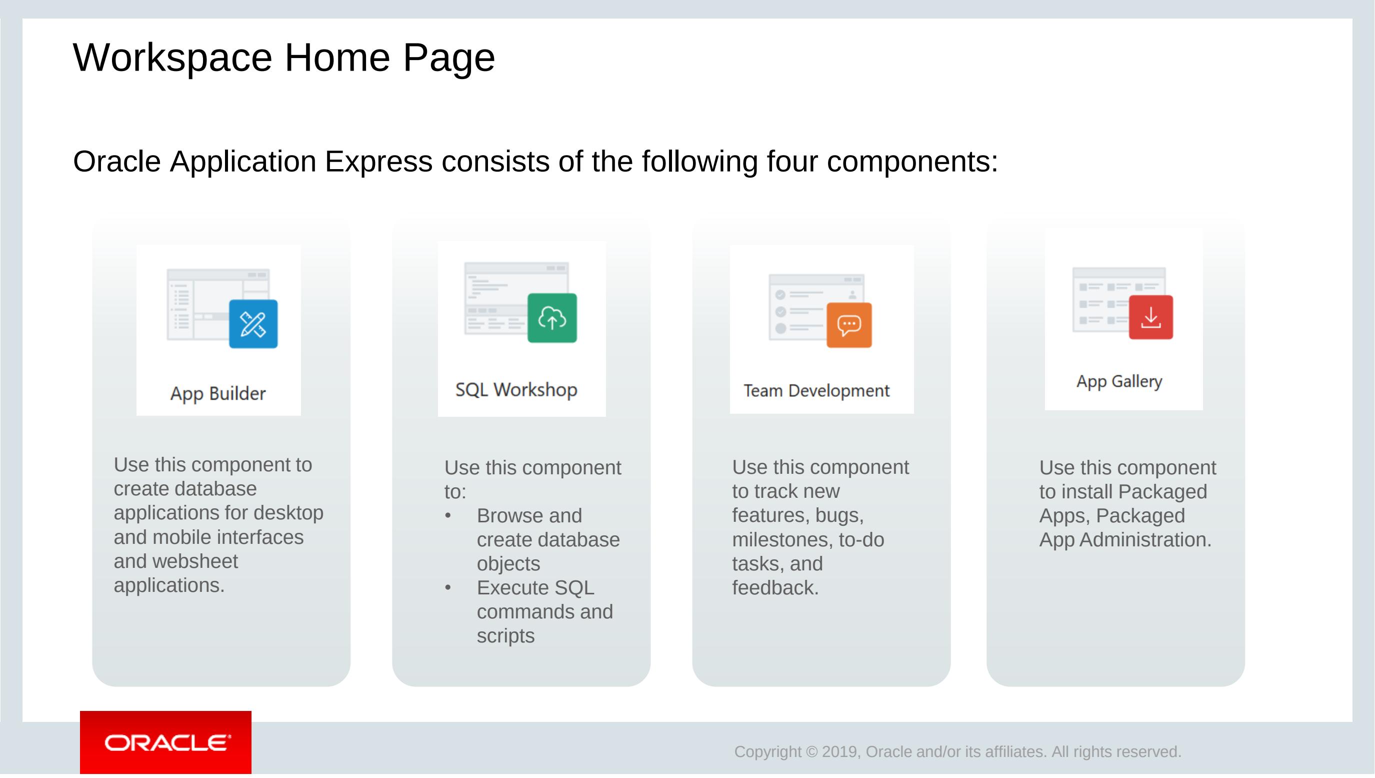
You can also view the Oracle Application Express development environment in a darker color theme. To switch to the Dark Mode theme, click the **Account Menu** icon in the top-right corner on your page and then click **Dark Mode.** The color theme immediately changes to a darker color theme as shown in the screenshot. To revert to the original mode, click the Account Menu icon and click **Dark Mode** again.



Practice 2-1 Overview: Using Oracle Application Express as a Workspace Administrator

The practices for this lesson cover the following topics:

- Logging in to a workspace
- Creating a developer user



Workspace Home Page

Oracle Application Express consists of the following four components:

App Builder

Use this component to create database applications for desktop and mobile interfaces and websheet applications.

SQL Workshop

Use this component to:
- Browse and create database objects
- Execute SQL commands and scripts

Team Development

Use this component to track new features, bugs, milestones, to-do tasks, and feedback.

App Gallery

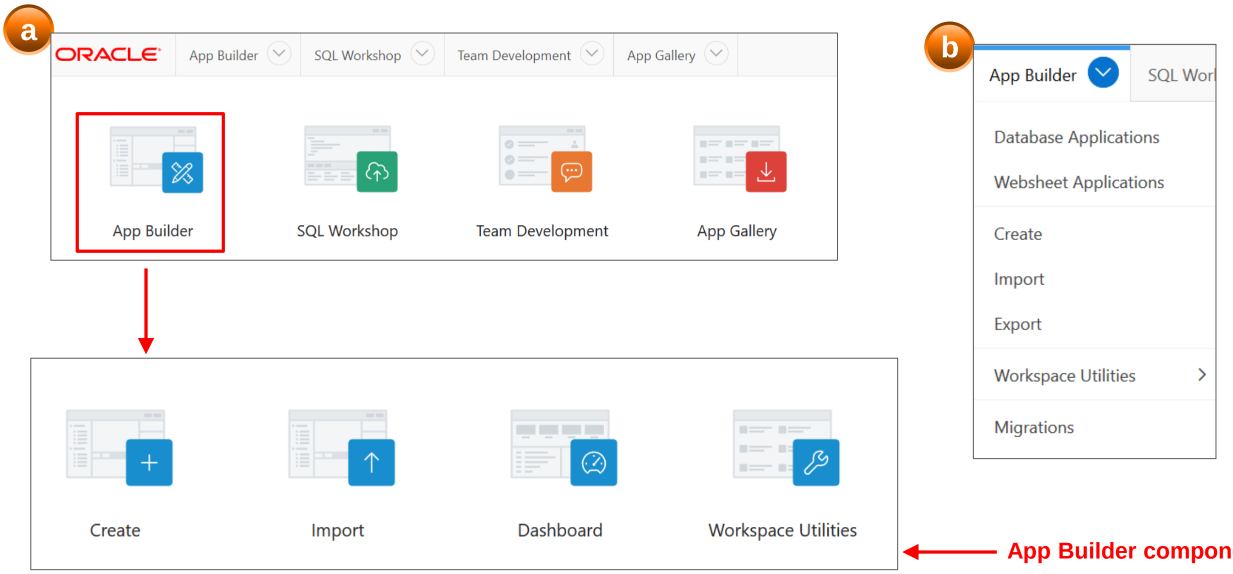
Use this component to install Packaged Apps, Packaged App Administration.

When you sign in to the Oracle Application Express development environment, you have to access any one of the following four components that you see on the Workspace home page:

- **App Builder:** To create an application, composed of a set of HTML pages, based on database objects. You can create application pages and use the built-in features to add reports, forms, charts, calendars, and so on to an application. Using Application Builder, you can build database applications and websheet applications. You learn to create a database application in the lesson titled “Creating a Database Application.”
- **SQL Workshop:** To access and manage database objects of an application. You can browse the objects in your application schema. You can create database objects, such as tables, views, sequences, and so on. You can execute SQL commands and run SQL scripts.
- **Team Development:** To access the development management tool and track new features, bugs, milestones, to-do tasks, and feedbacks
- **App Gallery:** To install and unlock Packaged Apps. It provides a gallery of Packaged Apps that can be used in real time with minimum modification along with Sample Apps to give overview of Oracle Application Express features.

What Is App Builder?

The First Component of Oracle Application Express



App Builder components

The first component of Oracle Application Express is the **App Builder** tool, using which you can build an application. You can access the App Builder in two ways. On the Oracle Application Express home page:

- Click the **App Builder** icon and then select the component that you want to access.
- Click the down arrow on **the App Builder** tab and then select the component that you want to access in the drop-down menu.

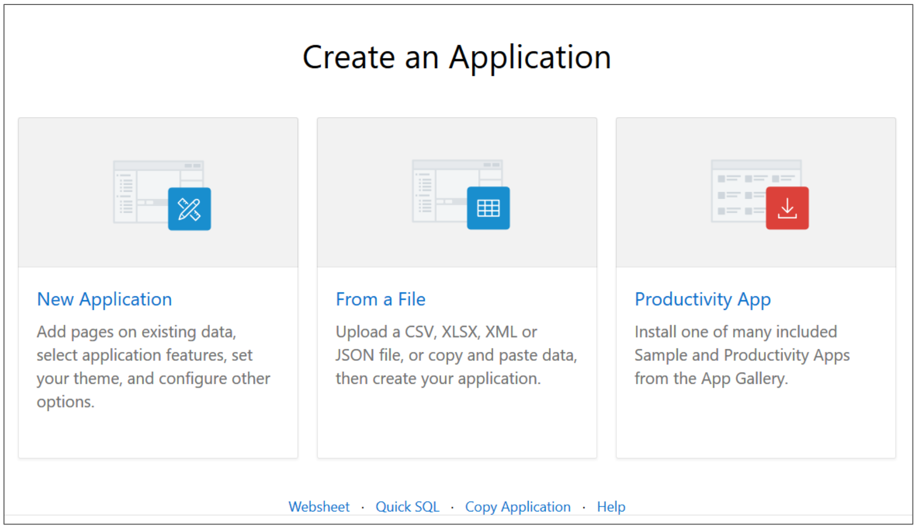
App Builder consists of the following four components:

- **Create:** Enables you to create database applications or websheet applications. It also allows you to install any packaged application in your workspace.
- **Import:** Enables you to import applications, Plug-Ins, Themes, User Interface Defaults, and Team Development Feedback into the workspace
- **Dashboard:** Provides details of all applications present in this workspace
- **Workspace Utilities:** Enables you to manage various attributes of the workspace

Note: When you sign in to Oracle Application Express and select **App Builder,** you will find that a Sample Database Application, which is a packaged application, is already installed for you.

Types of Applications

- New Application (Database)
- From a Spreadsheet
- Productivity App
- Websheet
- Quick SQL
- Copy Application



Create an Application

New Application
Add pages on existing data, select application features, set your theme, and configure other options.

From a File
Upload a CSV, XLSX, XML or JSON file, or copy and paste data, then create your application.

Productivity App
Install one of many included Sample and Productivity Apps from the App Gallery.

Websheet · Quick SQL · Copy Application · Help

When you click **Create** in App Builder, it opens the **Create Application Wizard,** which prompts you to choose the type of application that you want to create. Options include:

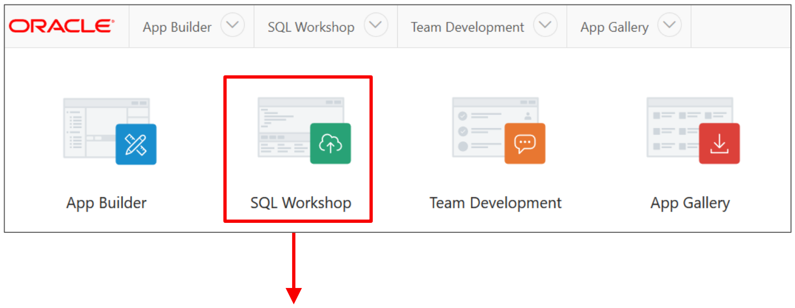
- **New Application:** A database application is a collection of pages that share a common session state and authentication. Database applications enable you to have full control on all aspects of the development process. With database applications, you can directly leverage your SQL and PL/SQL programming skills. You can manually add and customize components such as reports, charts, or forms, page controls such as buttons, items, or lists of values, and Shared Components that includes breadcrumbs, lists, or tabs.
  Steve decides to use the option **New Application** in the wizard to create the PTS application.
- **From a File:** You can also create an application by loading data from a file as comma separated (*.csv) or tab delimited file or by copying and pasting. You can then create an interactive report on that data.
- **Productivity App:** Productivity Apps are fully functional applications that are designed to address a specific business need.
- **Websheet:** A websheet application is geared toward the business user and requires no prior development experience. Each websheet application is a collection of pages designed for web-based data entry and reporting. Websheet applications are simplified and support pages, data grids, and reports. When you create a websheet application, **App Builder** automatically handles the creation of tables, triggers, and sequences.

For more details about websheet applications and other options in the wizard, see *“Oracle Application Express App Builder User’s Guide.”*

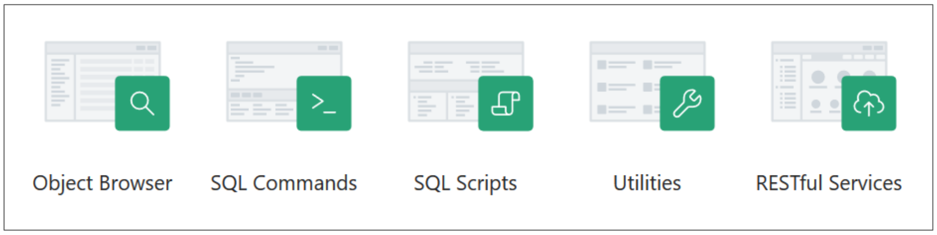
What Is SQL Workshop?

The Second Component of Oracle Application Express

- Create, view, and edit database objects.
- Load and unload data.
- Generate DDL statements.



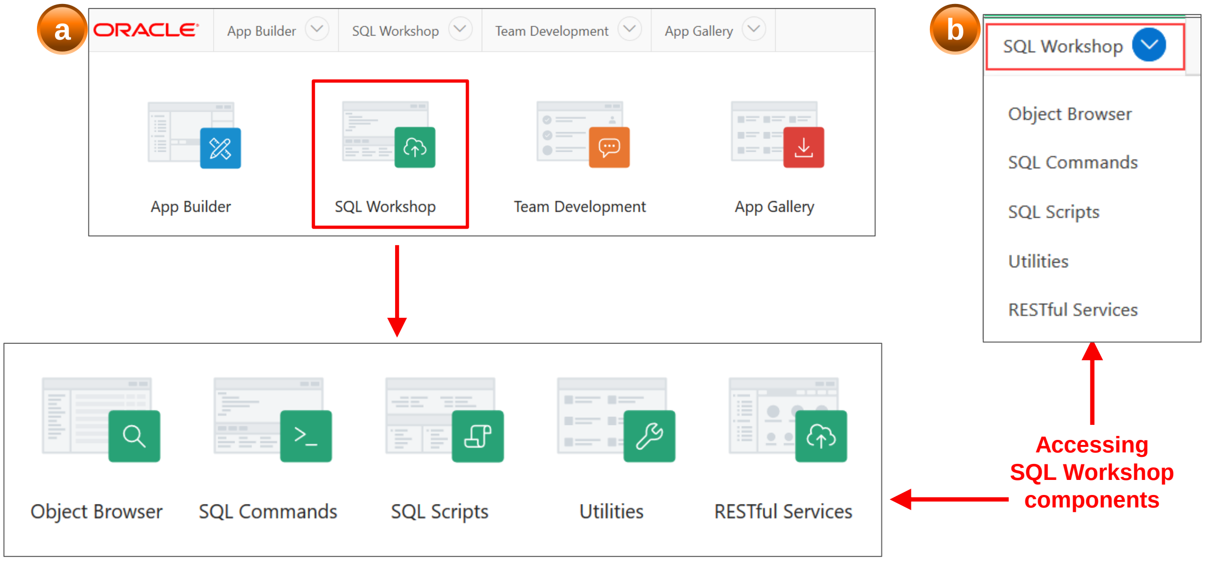
SQL Workshop components



The second component of Oracle Application Express is the **SQL Workshop** tool in Oracle Application. You can use it to interact with database objects. You can create, view, and edit database objects. You can also perform tasks such as loading data to and unloading data from database tables, generating data definition language (DDL) statements, and viewing reports. SQL Workshop consists of the following five components:

- **Object Browser:** Enables you to browse, create, and edit objects in a database
- **SQL Commands:** Enables you to create, edit, view, run, and delete database objects
- **SQL Scripts:** Is a set of SQL commands saved as a file in SQL Scripts. A SQL script can contain one or more SQL statements or PL/SQL blocks. You can use SQL Scripts to create, edit, view, run, and delete database objects.
- **Utilities:** Enables you to build SQL queries, load and unload data from an Oracle database, generate DDL, view object reports, manage User Interface defaults, restore dropped database objects, compare schemas, monitor the database, and view database details
- **RESTful Services:** Enables the declarative specification of RESTful Web Services used to access the database. These services work in conjunction with the Oracle Application Express Listener to enable the consumption of these services.

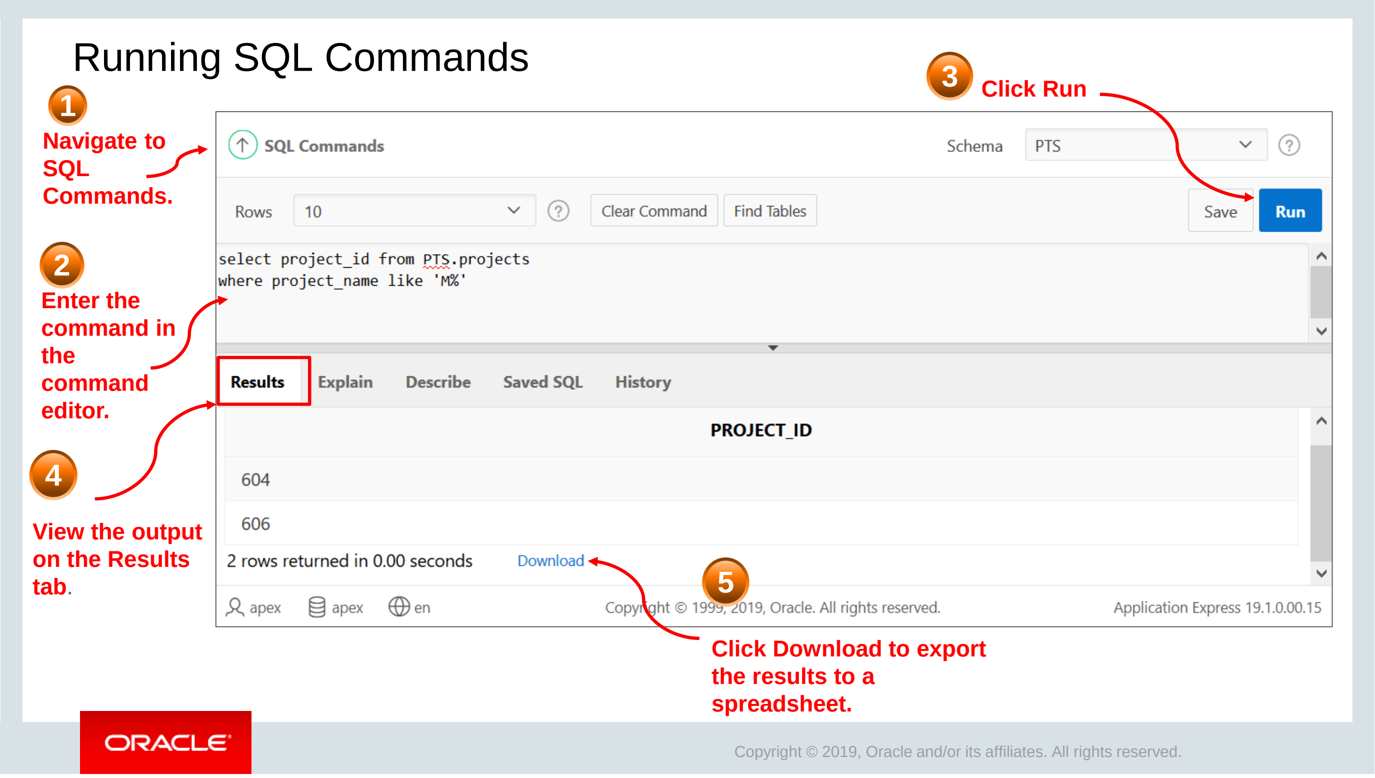
Accessing SQL Workshop



Accessing SQL Workshop components

From the Oracle Application Express home page, you can access the **SQL Workshop** tool in two ways:

- Click the **SQL Workshop** icon or the **SQL Workshop** tab and select the component that you want to access.
- Click the down arrow on the SQL Workshop tab and select the component that you want to access from the drop-down menu.



Running SQL Commands

1 Navigate to SQL Commands.

2 Enter the command in the command editor.

3 Click Run

4 View the output on the Results tab.

5 Click Download to export the results to a spreadsheet.

```
select project_id from PTS.projects
where project_name like 'M%'
```

PROJECT_ID
604
606

2 rows returned in 0.00 seconds Download

apex apex en Copyright © 1999, 2019, Oracle. All rights reserved. Application Express 19.1.0.00.15

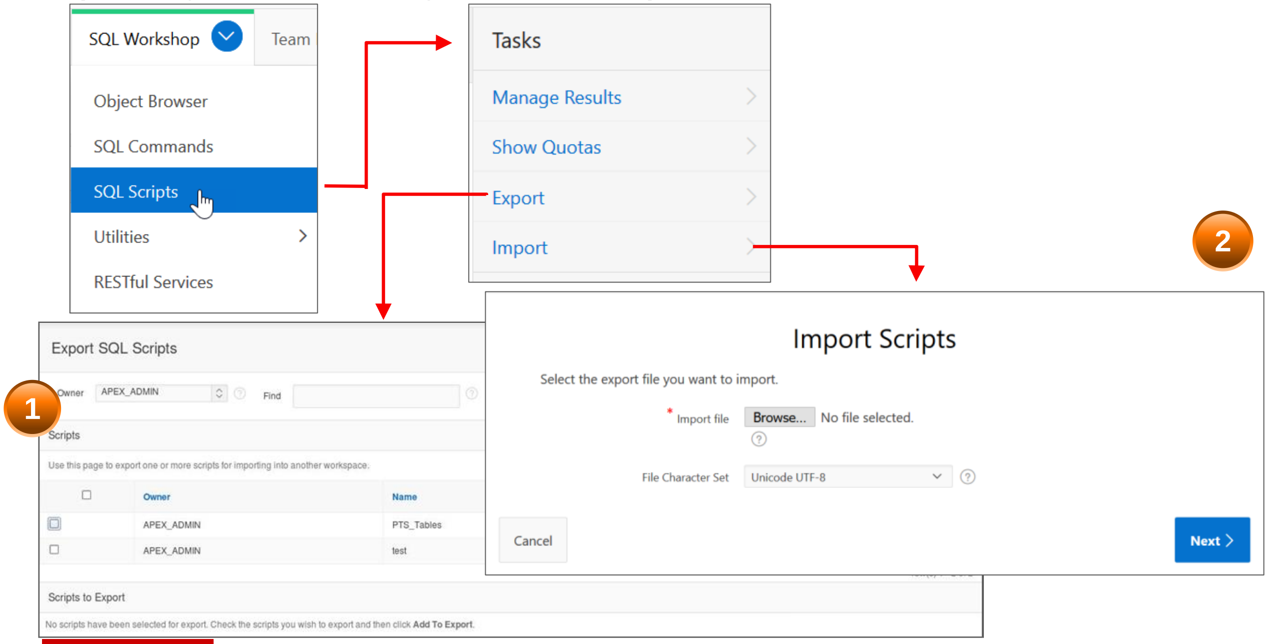
In the SQL Command component of SQL Workshop, you can execute SQL scripts. To execute SQL code with SQL Commands, perform the following steps:

1. Navigate to the SQL Commands page by selecting **SQL Commands** from the drop-down menu on the SQL Workshop tab.
2. Enter the SQL or PL/SQL statement in the command editor.
3. Click the **Run** button.
4. View the output on the Results tab of the display pane.
5. Optionally, click the **Download** link to export the results of the query to a spreadsheet in Microsoft Excel.

Notes

- If you have multiple commands in the command editor, you can run only one command at a time. Select the command and click **Run.** Only the command that was selected is executed.
- SQL commands that are created and saved by using Query Builder can be executed from the SQL Commands page.

Importing and Running a SQL Script



Owner: APEX_ADMIN

Use this page to export one or more scripts for importing into another workspace.

☐	Owner	Name
☐	APEX_ADMIN	PTS_Tables
☐	APEX_ADMIN	test

No scripts have been selected for export. Check the scripts you wish to export and then click **Add To Export**.

Select the export file you want to import.

* Import file: Browse... No file selected.

File Character Set: Unicode UTF-8

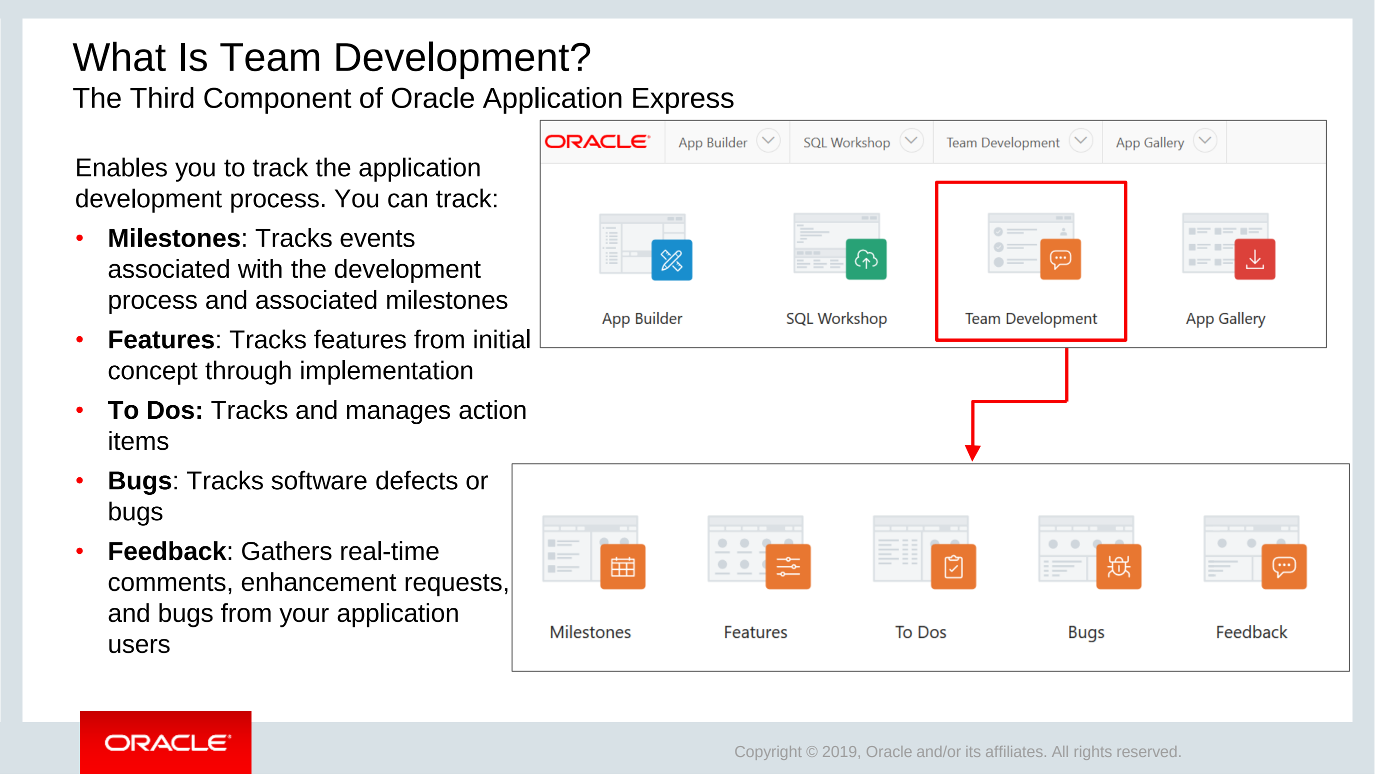
The option to export and import SQL scripts between workspaces, **SQL Scripts**, is available under the SQL Workshop component**.**

Export Scripts: By using this option, you can export multiple scripts from your current workspace to another workspace. All the scripts that you select to export are encoded into a single script file. You can save this file to your local file system and import it to another workspace. To export scripts:

1. Go to **SQL Workshop** and select **SQL Scripts** and **Export link** under Tasks. The scripts available in the script repository are listed in the Scripts pane.
2. Select the scripts that you want to export and click the **Add To Export** button. The selected scripts are listed in the “Scripts to Export” pane. You can finalize the scripts that you want to export by removing or adding scripts. To export all the scripts, click the **Export All** button. The scripts are exported as a single export file, which you can save to your local file system.

Import Scripts: By using this option, you can import a script file exported from a different workspace into your current workspace. To import a script file:

1. Go to **SQL Workshop** and select **SQL Scripts** and **Import link** under Tasks.
2. Click the Browse button and locate the file to import from your local file system.
3. Click **Next.**
4. Click **Import Scripts** to confirm. Only the script files exported from the scripts repository can be imported. If you try to import any other script, you get a “script not compatible” error.



What Is Team Development?

The Third Component of Oracle Application Express

Enables you to track the application development process. You can track:

- **Milestones**: Tracks events associated with the development process and associated milestones
- **Features**: Tracks features from initial concept through implementation
- **To Dos:** Tracks and manages action items
- **Bugs**: Tracks software defects or bugs
- **Feedback**: Gathers real-time comments, enhancement requests, and bugs from your application users

The third component of Oracle Application Express is the **Team Development** tool in Oracle Application. By using the components of this tool, you can track the following in an application development process:

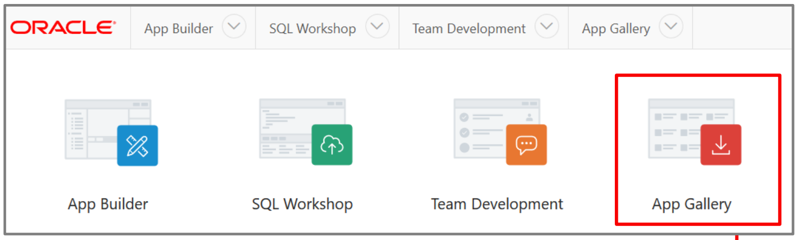
- Milestones
- Features
- To Dos
- Bugs
- Feedback

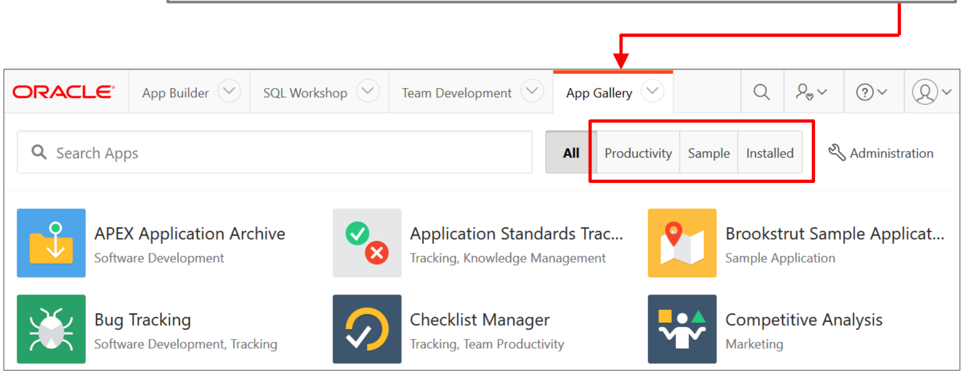
What Is App Gallery?

The Fourth Component of Oracle Application Express

App Gallery includes two types of applications:

- Productivity applications
- Sample applications





ORACLE

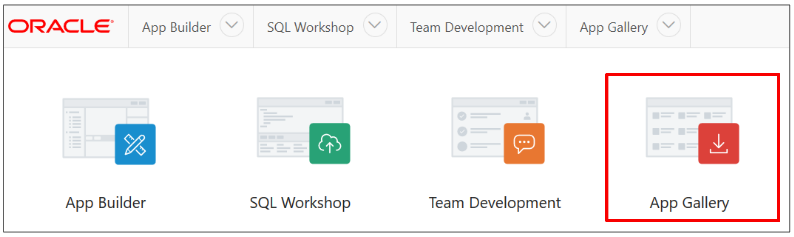
The fourth component of Oracle Application Express is the **App Gallery** in Oracle Application. The App Gallery provides two types of applications:

- **Productivity applications:** These are fully functional applications designed for specific business needs. You can install, run, and use a productivity application as is. You can analyze its technical and functional logic to build a specific type of functionality. Productivity applications are not editable by default. You must unlock the Productivity applications before you can edit them. Unlocking an application makes it ineligible for future upgrades or support by Oracle Support.
- **Sample Applications:** Sample Applications are not complete applications in itself but contain code snippets or sample codes. Sample applications are available for installation in a workspace and are editable by default.

Accessing the App Gallery

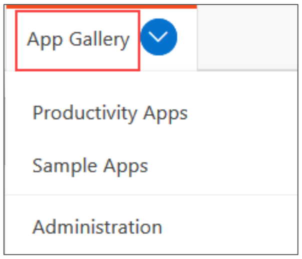
1

App Gallery on the Workspace home page



2

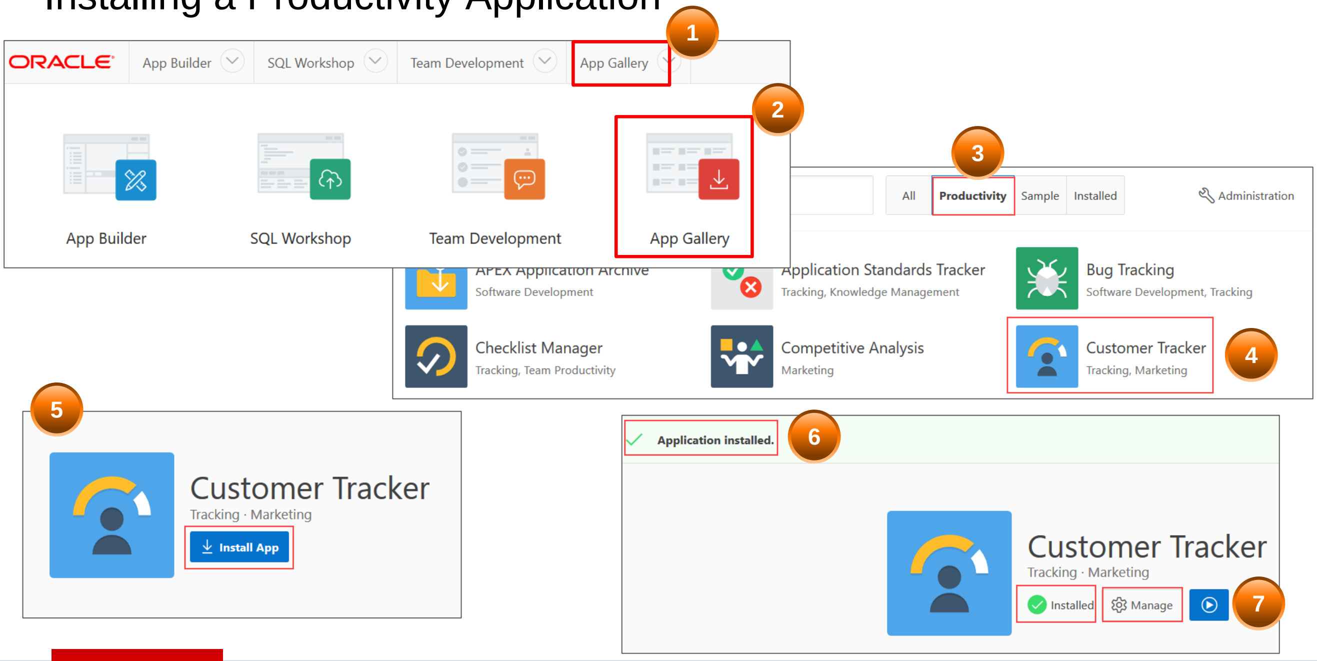
App Gallery drop-down list



To access the App Gallery, log in to Oracle Application Express. On the Workspace home page:

- Click **App Gallery** icon as shown in screenshot a.
- Click the **App Gallery** tab to open the App Gallery drop-down list, as shown in screenshot b.

Installing a Productivity Application



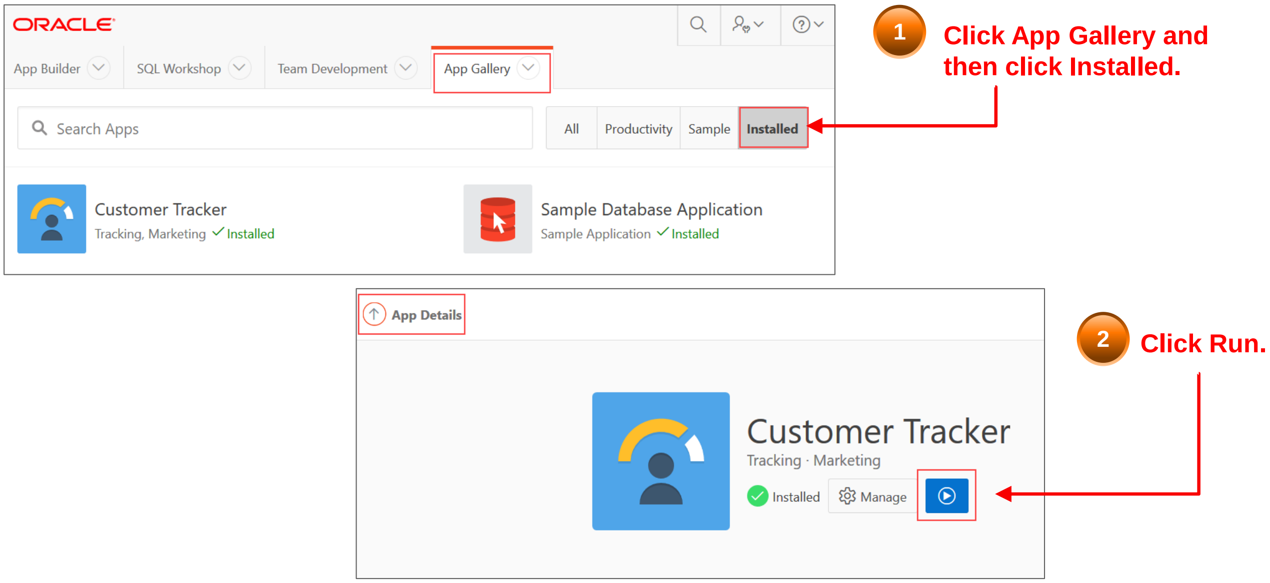
ORACLE

To install a Productivity application:

1. On the App Builder home page, click the **App Gallery** tab or the **App Gallery** icon. The App Gallery page opens.
2. On the App Gallery page, click **Productivity.**
3. On the Productivity App page, locate the application to be installed. In the example slide, the **Customer Tracker** application is selected.
4. Click the application icon. The application summary page opens. Click **Install App.**
5. Select an Authentication scheme and click **Next.**
6. Click **Install Application** again. A success message appears.
7. To run the application, click the **Run** icon.
8. Enter the appropriate login credentials and click **Sign In.**

   The application is installed.

Running an Installed Productivity Application

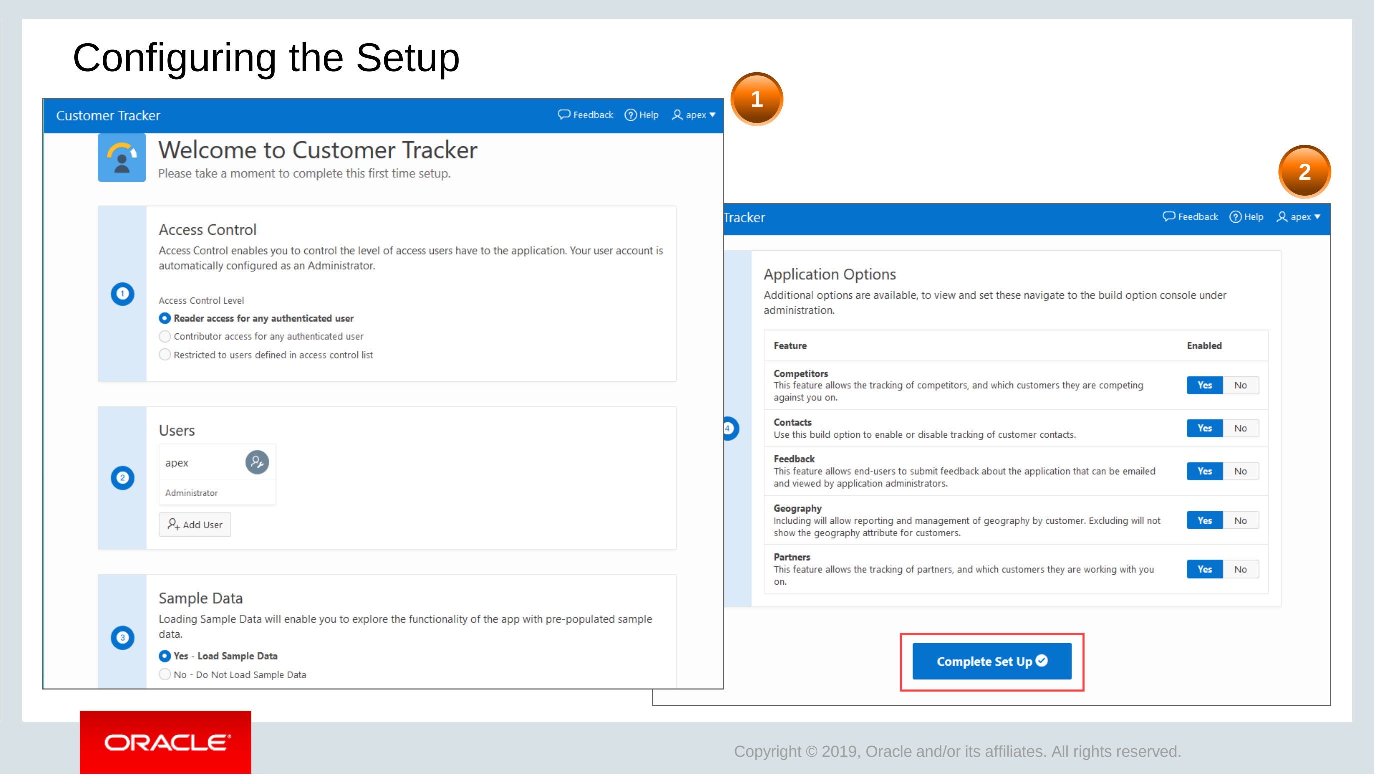


1 Click App Gallery and then click Installed.

2 Click Run.

To run an installed Productivity application:

1. Click the **App Gallery** tab. The App Gallery page appears.
2. Click **Installed.** Locate the application.
3. Click the application that you want to run. The App Details page opens.
4. Click the **Run** icon.
5. Enter the appropriate login credentials to view the application.



Configuring the Setup

Customer Tracker — Feedback · Help · apex

Welcome to Customer Tracker

Please take a moment to complete this first time setup.

1 Access Control

Access Control enables you to control the level of access users have to the application. Your user account is automatically configured as an Administrator.

Access Control Level

● **Reader access for any authenticated user**
○ Contributor access for any authenticated user
○ Restricted to users defined in access control list

2 Users

apex — Administrator

Add User

3 Sample Data

Loading Sample Data will enable you to explore the functionality of the app with pre-populated sample data.

● **Yes - Load Sample Data**
○ No - Do Not Load Sample Data

Tracker — Feedback · Help · apex

4 Application Options

Additional options are available, to view and set these navigate to the build option console under administration.

Feature	Enabled
Competitors This feature allows the tracking of competitors, and which customers they are competing against you on.	**Yes** No
Contacts Use this build option to enable or disable tracking of customer contacts.	**Yes** No
Feedback This feature allows end-users to submit feedback about the application that can be emailed and viewed by application administrators.	**Yes** No
Geography Including will allow reporting and management of geography by customer. Excluding will not show the geography attribute for customers.	**Yes** No
Partners This feature allows the tracking of partners, and which customers they are working with you on.	**Yes** No

Complete Set Up ✓

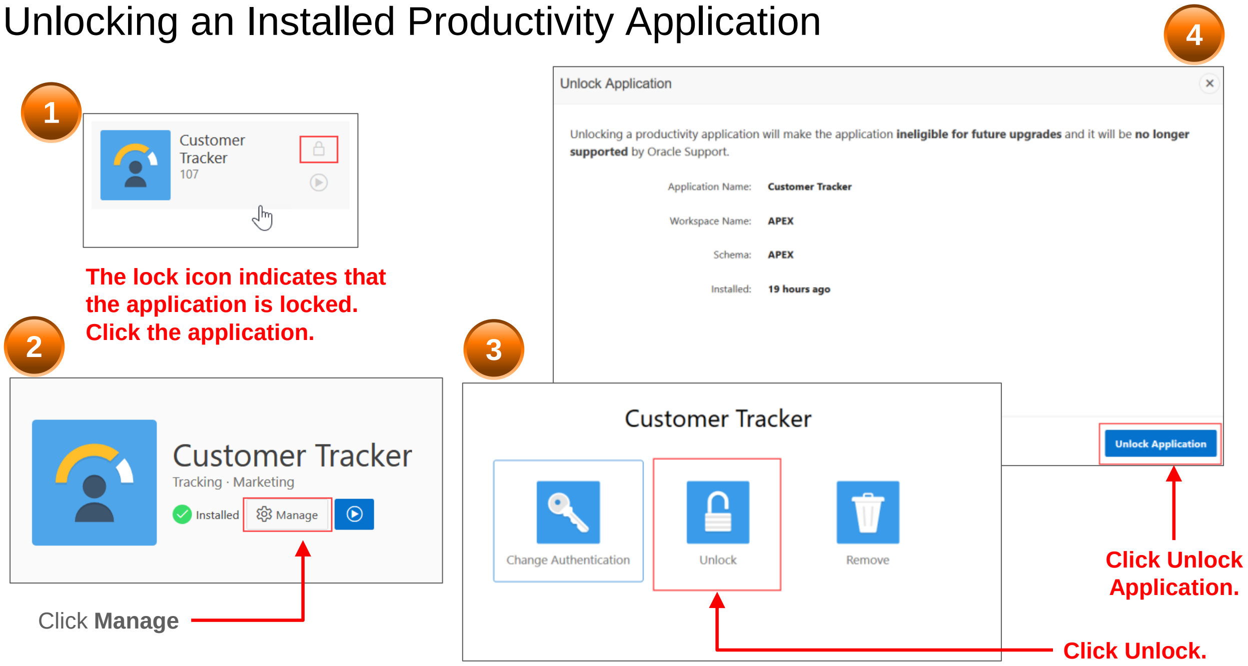
ORACLE

The setup page opens when you run the application for the first time. In this example, the **Customer Tracker** application is considered for first-time setup. You can set up the following:

- **Access Control** – Assign different access levels for authenticated users such as reader access, contributor access, or restricted access.
- **Users** – Add users and assign roles such as contributor, reader, and administrator to the users.
- **Sample Data** – Load sample data.
- **Application Options** – Enable or disable features such as contacts, feedback, and so on.

After you have completed setting up the features, click **Complete Set Up.** The Customer Tracker application opens.

Unlocking an Installed Productivity Application



Unlock Application

Unlocking a productivity application will make the application **ineligible for future upgrades** and it will be **no longer supported** by Oracle Support.

Application Name: **Customer Tracker**

Workspace Name: **APEX**

Schema: **APEX**

Installed: **19 hours ago**

The lock icon indicates that the application is locked. Click the application.

Click **Manage**

Click Unlock.

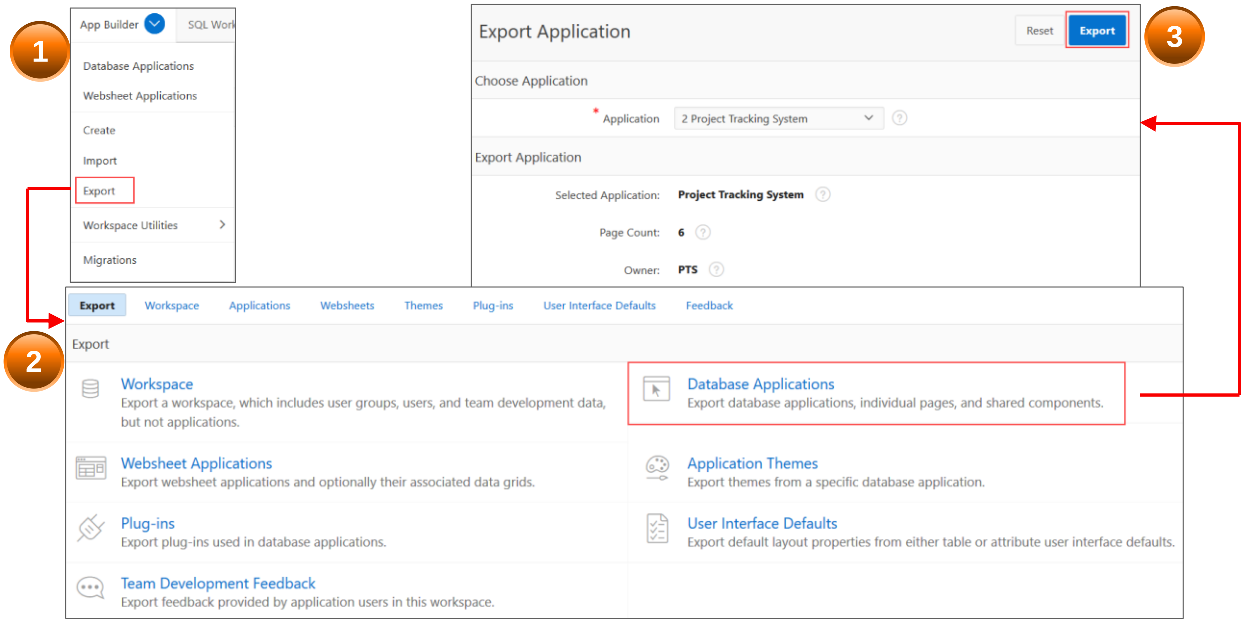
Click Unlock Application.

ORACLE

After a Productivity application is installed, you must unlock it before you can edit it. In this example, the **Customer Tracker** app is considered for unlocking. To unlock the application:

1. Open the **App Gallery** home page and click the app to be unlocked.
2. On the App Details page, click **Manage.**
3. Select **Unlock.** The Unlock Application dialog box opens.
4. In the Unlock Application dialog box, click **Unlock Application.**

Exporting an Application


1
App Builder
SQL Work
Database Applications
Websheet Applications
Create
Import
Export
Workspace Utilities
Migrations
2
Export
Workspace
Applications
Websheets
Themes
Plug-ins
User Interface Defaults
Feedback
Export
Workspace
Export a workspace, which includes user groups, users, and team development data, but not applications.
Database Applications
Export database applications, individual pages, and shared components.
Websheet Applications
Export websheet applications and optionally their associated data grids.
Application Themes
Export themes from a specific database application.
Plug-ins
Export plug-ins used in database applications.
User Interface Defaults
Export default layout properties from either table or attribute user interface defaults.
Team Development Feedback
Export feedback provided by application users in this workspace.
3
Export Application
Reset
Export
Choose Application
Export Application

* Application: 2 Project Tracking System

Selected Application: **Project Tracking System**

Page Count: **6**

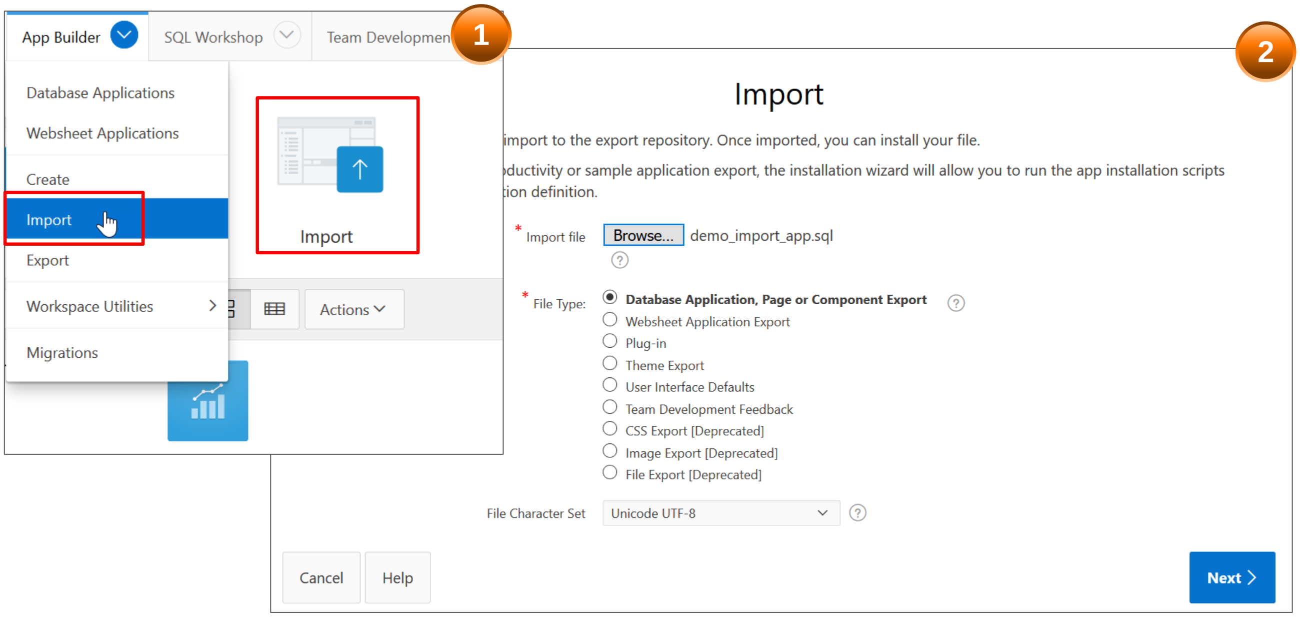
Owner: **PTS**

You may want to take a backup of your application or move it to a different workspace. You can do this by exporting your application. When you export an application, the application definitions and all associated files are exported.

To export an application:

1. From the App Builder drop-down list, select **Export.**
2. On the Export page, select **Database Applications.** The Export Application page opens.
3. From the **Application** drop-down list, select the application that you want to export. In this example, the Project Tracking System is selected.
4. Choose the appropriate options and enter required details, and click **Export** to export an application.

Importing an Application



Import

import to the export repository. Once imported, you can install your file.

oductivity or sample application export, the installation wizard will allow you to run the app installation scripts tion definition.

* Import file Browse... demo_import_app.sql

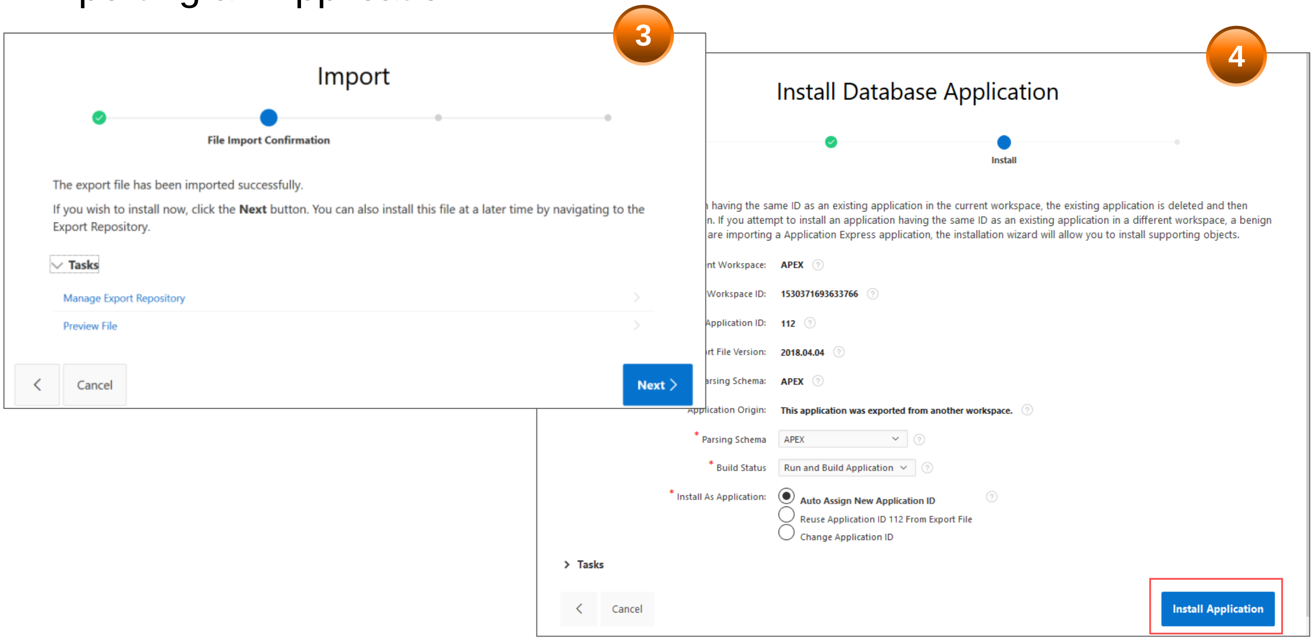
* File Type:
● Database Application, Page or Component Export
○ Websheet Application Export
○ Plug-in
○ Theme Export
○ User Interface Defaults
○ Team Development Feedback
○ CSS Export [Deprecated]
○ Image Export [Deprecated]
○ File Export [Deprecated]

File Character Set Unicode UTF-8

After you export an application, you can import them into the target Oracle Application Express instance and then install it. As a general rule, always import the application first and then the related files. To import an Application, Page, or Component Export into a target Oracle Application Express instance:

1. Navigate to the Import page:
   - From the App Builder drop-down list, select **Import.**
   - Click the **Import** icon on the App Builder home page.
2. On the Import page, click **Browse** to navigate to the file and select the file type. Verify that the File Character Set is correct.
3. Click **Next.** After you import a file, you have the option to install it.

Importing an Application



Import

File Import Confirmation

The export file has been imported successfully.

If you wish to install now, click the **Next** button. You can also install this file at a later time by navigating to the Export Repository.

Install Database Application

Install

h having the same ID as an existing application in the current workspace, the existing application is deleted and then
n. If you attempt to install an application having the same ID as an existing application in a different workspace, a benign
are importing a Application Express application, the installation wizard will allow you to install supporting objects.

nt Workspace: **APEX**

Workspace ID: **1530371693633766**

Application ID: **112**

rt File Version: **2018.04.04**

rsing Schema: **APEX**

Application Origin: **This application was exported from another workspace.**

* Parsing Schema: APEX

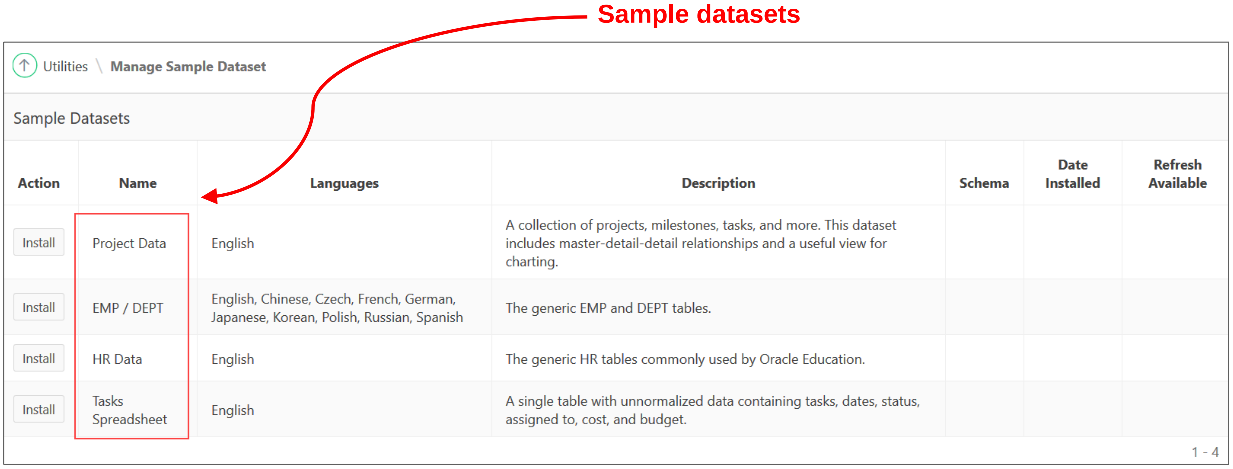
* Build Status: Run and Build Application

* Install As Application:
● **Auto Assign New Application ID**
○ Reuse Application ID 112 From Export File
○ Change Application ID

4. On File Import Confirmation, click **Next.** The Install Database Application page opens.
5. On the Install Database Application page, select the following:
   - **Parsing Schema:** Select the schema on which you want to install the application.
   - **Build Status:** You can select **Run Application Only** or **Run and Build Application.** In the Run Application Only status, users can only run an application.
     In the Run and Build Application status, users can run an application and developers can both run and edit an application.
   - **Install as Application:** Select an option to auto assign the application ID, reuse the application ID, or manually assign an application ID.
6. Click **Install Application** to install the application.

In case you import an application that is already existing in the workspace, it gets imported as a new application altogether with a different application ID. Importing or exporting an application does not impact the underlying data.

Sample Datasets



Sample datasets

Sample Datasets

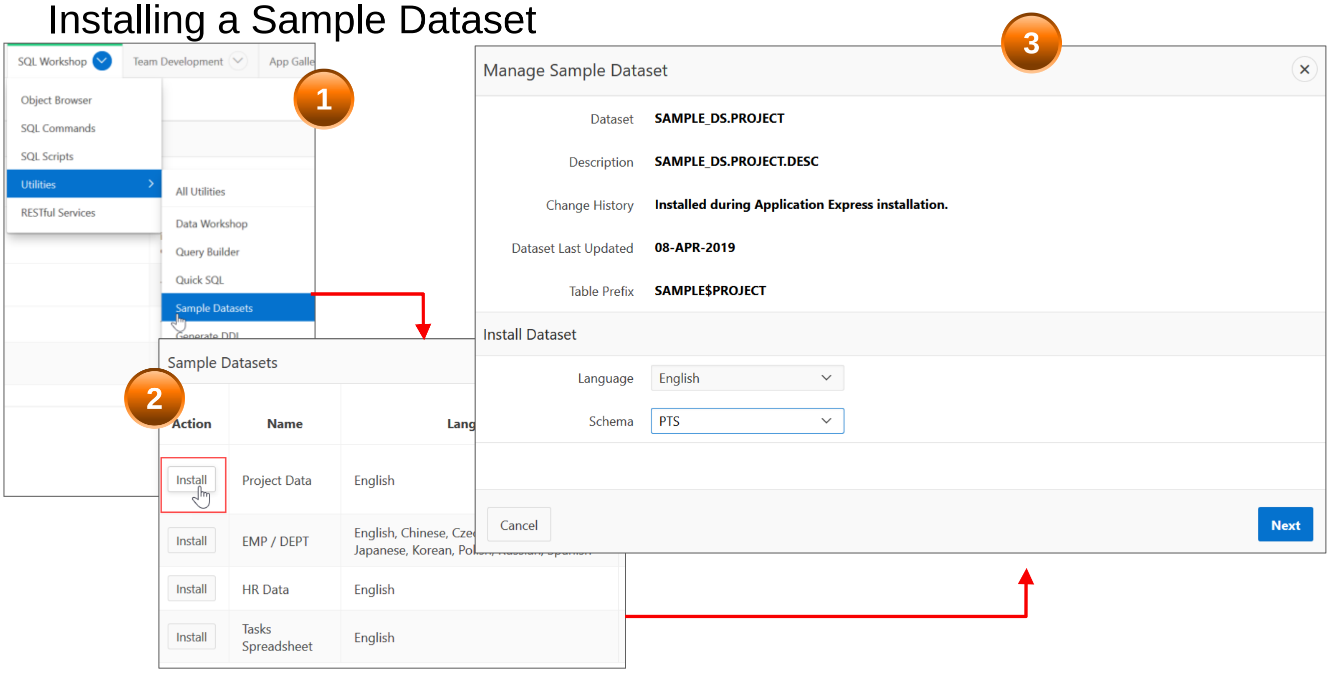
Action	Name	Languages	Description	Schema	Date Installed	Refresh Available
Install	Project Data	English	A collection of projects, milestones, tasks, and more. This dataset includes master-detail-detail relationships and a useful view for charting.			
Install	EMP / DEPT	English, Chinese, Czech, French, German, Japanese, Korean, Polish, Russian, Spanish	The generic EMP and DEPT tables.			
Install	HR Data	English	The generic HR tables commonly used by Oracle Education.			
Install	Tasks Spreadsheet	English	A single table with unnormalized data containing tasks, dates, status, assigned to, cost, and budget.			

1 - 4

Oracle Application Express includes four sample datasets – `Project Data, EMP/DEPT, HR Data,` and `Tasks Spreadsheet.` Each dataset includes sufficient data to enable you to build applications with pages containing different types of components.

With a few clicks of the mouse, you can install, refresh, replace, or remove sample datasets within one of the schemas associated with your workspace.

Installing a Sample Dataset



Sample Datasets

Action	Name	Lang
Install	Project Data	English
Install	EMP / DEPT	English, Chinese, Cze Japanese, Korean, Po
Install	HR Data	English
Install	Tasks Spreadsheet	English

Manage Sample Dataset

Dataset	SAMPLE_DS.PROJECT
Description	SAMPLE_DS.PROJECT.DESC
Change History	Installed during Application Express installation.
Dataset Last Updated	08-APR-2019
Table Prefix	SAMPLE$PROJECT

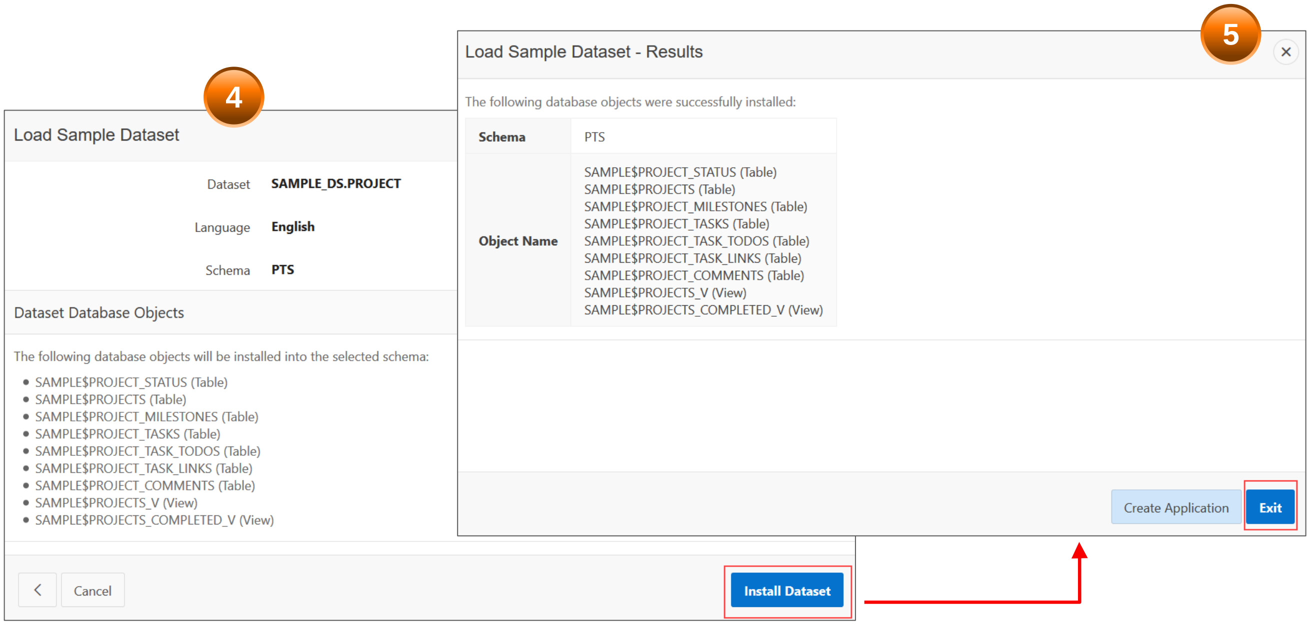
Install Dataset

Language	English
Schema	PTS

To install a sample dataset:

1. From the SQL Workshop drop-down list, select Utilities and click Sample Dataset (screenshot 1).
2. On the Sample Dataset page, select the sample dataset that you want to install and click Install. In this example, the *Project Data* sample dataset is selected (screenshot 2).
3. On the Manage Sample Dataset page, select the Schema on which you want to install the dataset and click Next. In this example, the PTS schema is selected (screenshot 3).

Installing a Sample Dataset



Load Sample Dataset

Dataset	SAMPLE_DS.PROJECT
Language	English
Schema	PTS

Dataset Database Objects

The following database objects will be installed into the selected schema:

- SAMPLE$PROJECT_STATUS (Table)
- SAMPLE$PROJECTS (Table)
- SAMPLE$PROJECT_MILESTONES (Table)
- SAMPLE$PROJECT_TASKS (Table)
- SAMPLE$PROJECT_TASK_TODOS (Table)
- SAMPLE$PROJECT_TASK_LINKS (Table)
- SAMPLE$PROJECT_COMMENTS (Table)
- SAMPLE$PROJECTS_V (View)
- SAMPLE$PROJECTS_COMPLETED_V (View)

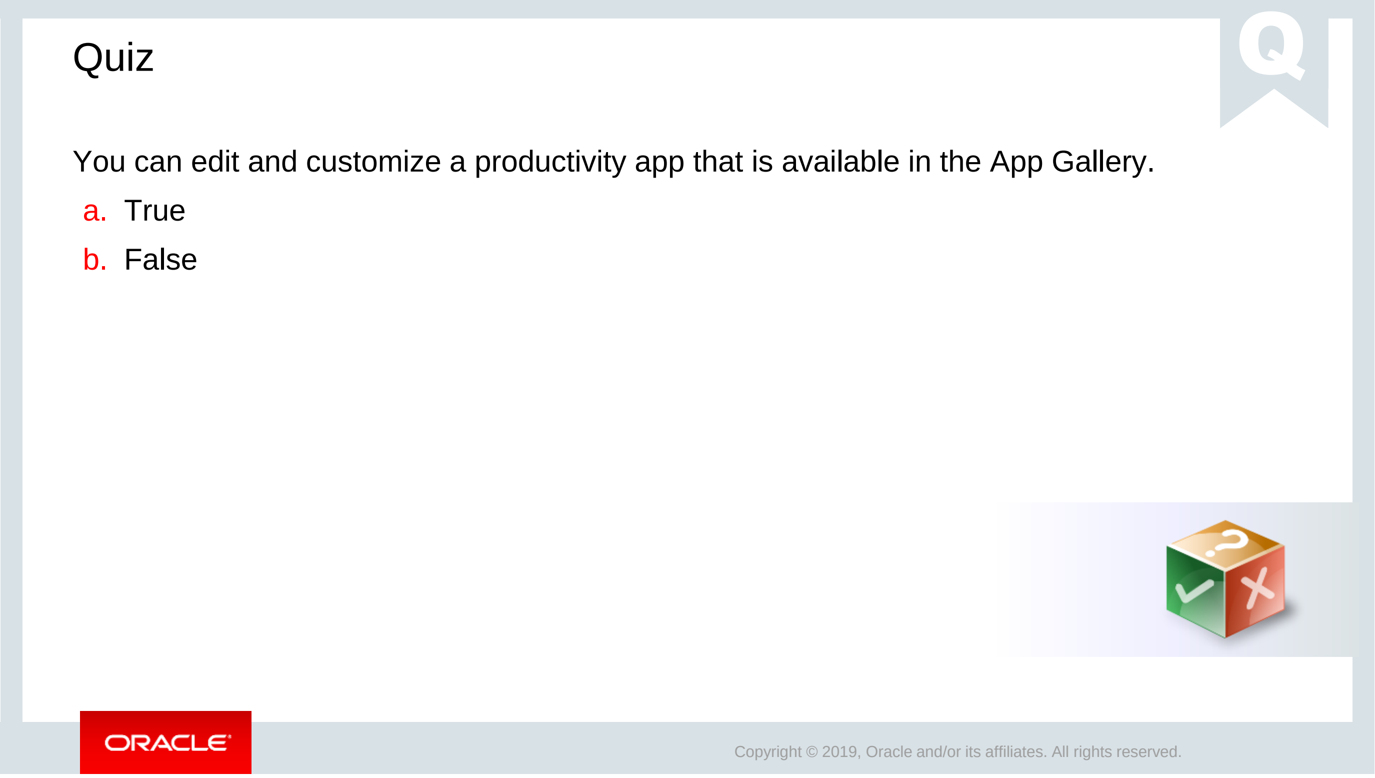
Load Sample Dataset - Results

The following database objects were successfully installed:

Schema	PTS
Object Name	SAMPLE$PROJECT_STATUS (Table) SAMPLE$PROJECTS (Table) SAMPLE$PROJECT_MILESTONES (Table) SAMPLE$PROJECT_TASKS (Table) SAMPLE$PROJECT_TASK_TODOS (Table) SAMPLE$PROJECT_TASK_LINKS (Table) SAMPLE$PROJECT_COMMENTS (Table) SAMPLE$PROJECTS_V (View) SAMPLE$PROJECTS_COMPLETED_V (View)

ORACLE

4. On the Load Sample Dataset page, the database objects that will be installed into the selected schema is listed. Click **Install Dataset** (screenshot 4).
5. After the sample dataset is successfully installed, you have the option to:
   - Create an application using the dataset set
   - Exit the wizard. Click **Exit** (screenshot 5).



Quiz

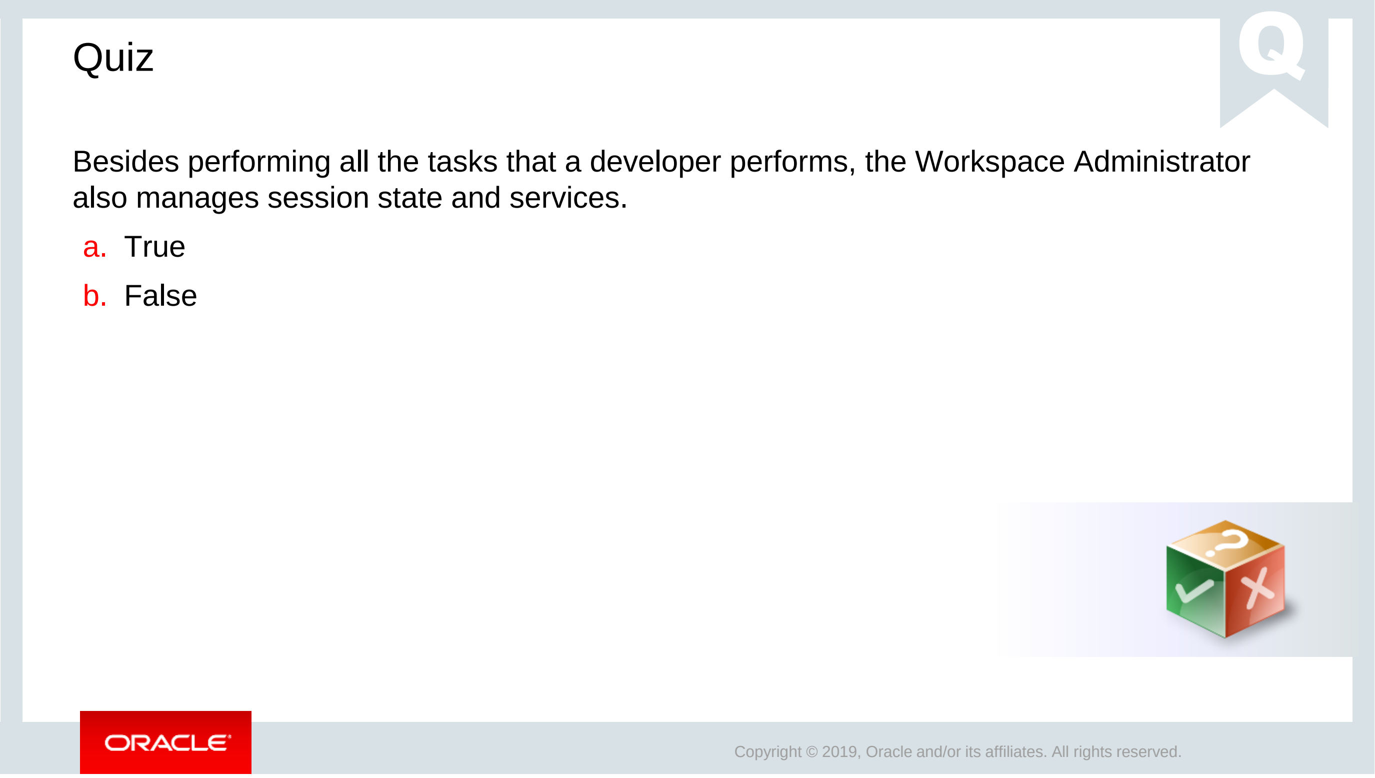
You can edit and customize a productivity app that is available in the App Gallery.

a. True

b. False

Answer: a

Note: You can edit or customize a productivity app if it has been unlocked.



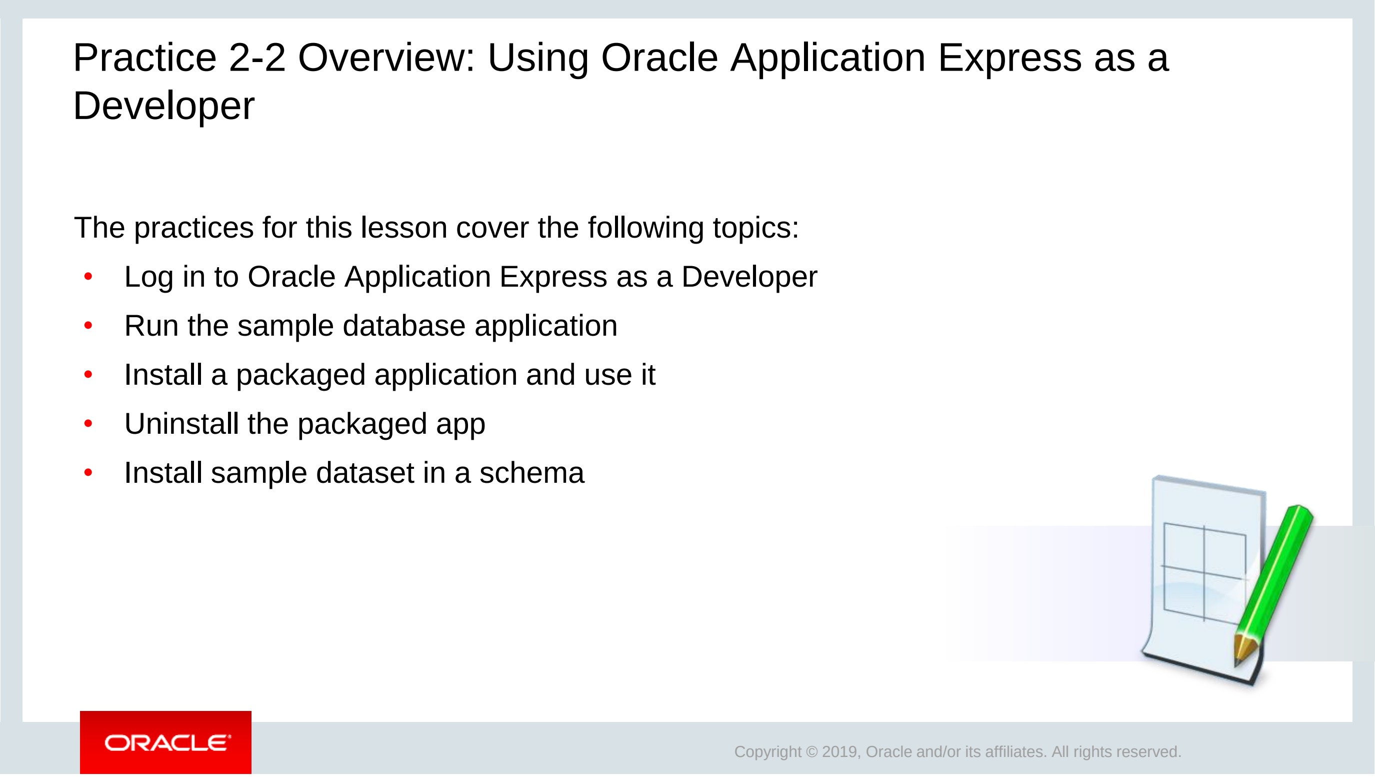
Quiz

Besides performing all the tasks that a developer performs, the Workspace Administrator also manages session state and services.

a. True

b. False

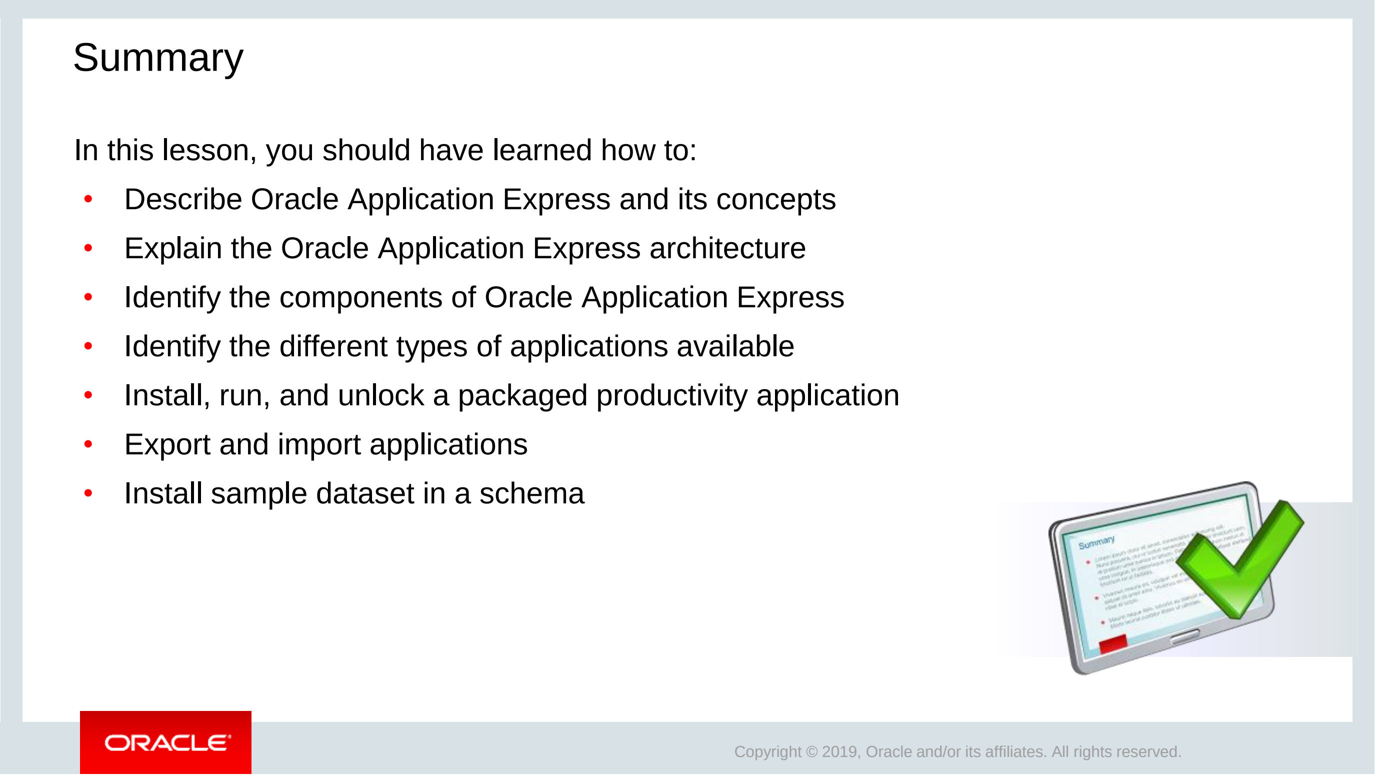
Answer: b



Practice 2-2 Overview: Using Oracle Application Express as a Developer

The practices for this lesson cover the following topics:

- Log in to Oracle Application Express as a Developer
- Run the sample database application
- Install a packaged application and use it
- Uninstall the packaged app
- Install sample dataset in a schema

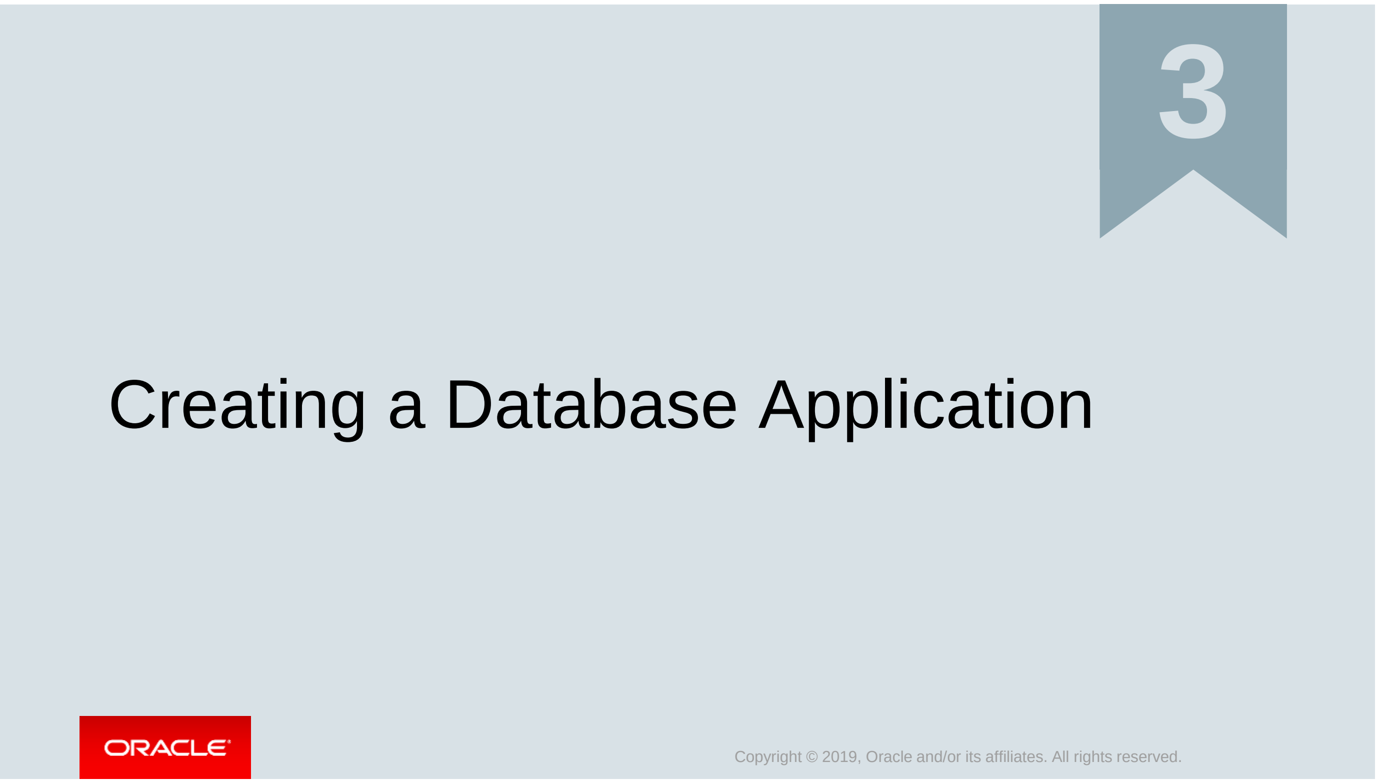


Summary

In this lesson, you should have learned how to:

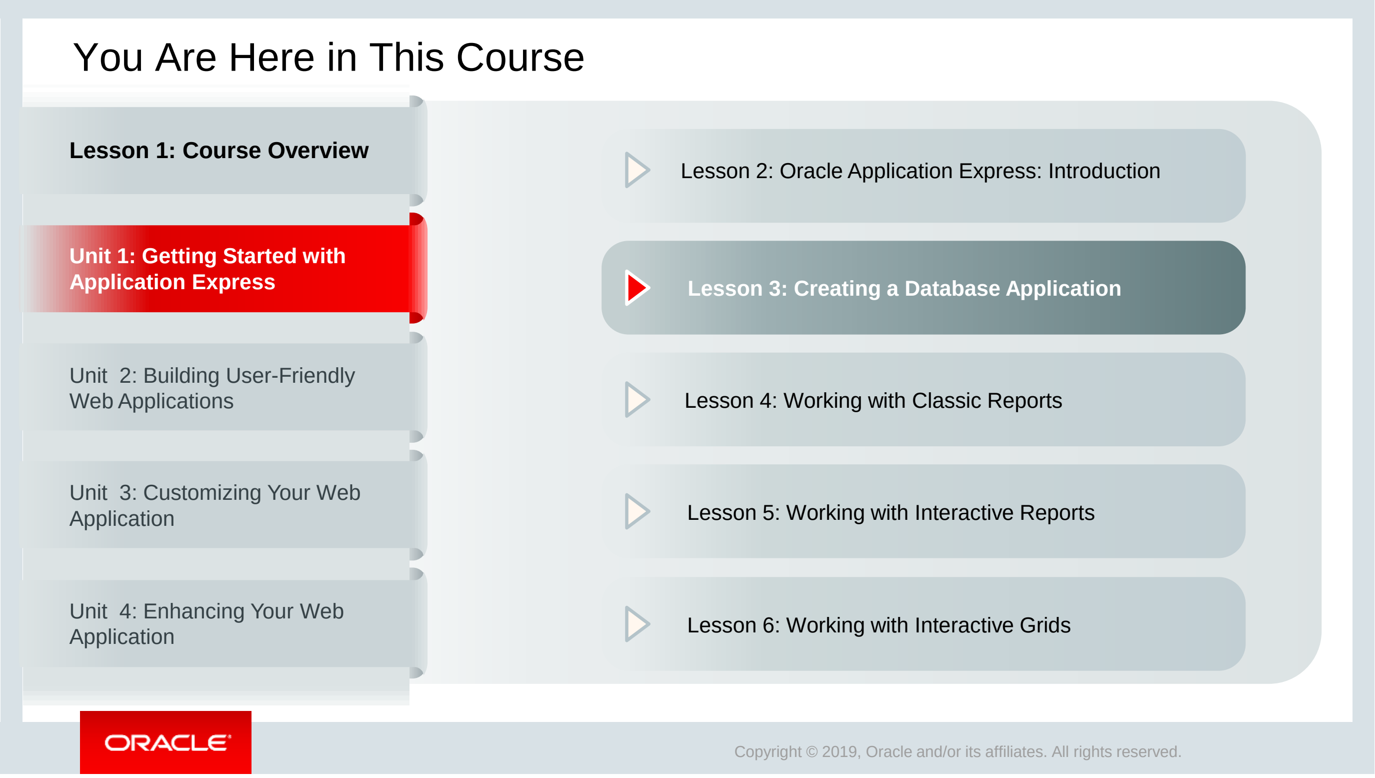
- Describe Oracle Application Express and its concepts
- Explain the Oracle Application Express architecture
- Identify the components of Oracle Application Express
- Identify the different types of applications available
- Install, run, and unlock a packaged productivity application
- Export and import applications
- Install sample dataset in a schema

Oracle Application Express is a rapid application development tool that is available in Oracle Database. In this lesson, you were introduced to Oracle Application Express, the advantages of using Oracle Application Express to build applications, and the Oracle Application Express features that you use when building your application. You also learned about the architecture and the components of Oracle Application Express, as well as the steps to get started.



3

Creating a Database Application



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- Lesson 2: Oracle Application Express: Introduction
- **Lesson 3: Creating a Database Application**
- Lesson 4: Working with Classic Reports
- Lesson 5: Working with Interactive Reports
- Lesson 6: Working with Interactive Grids

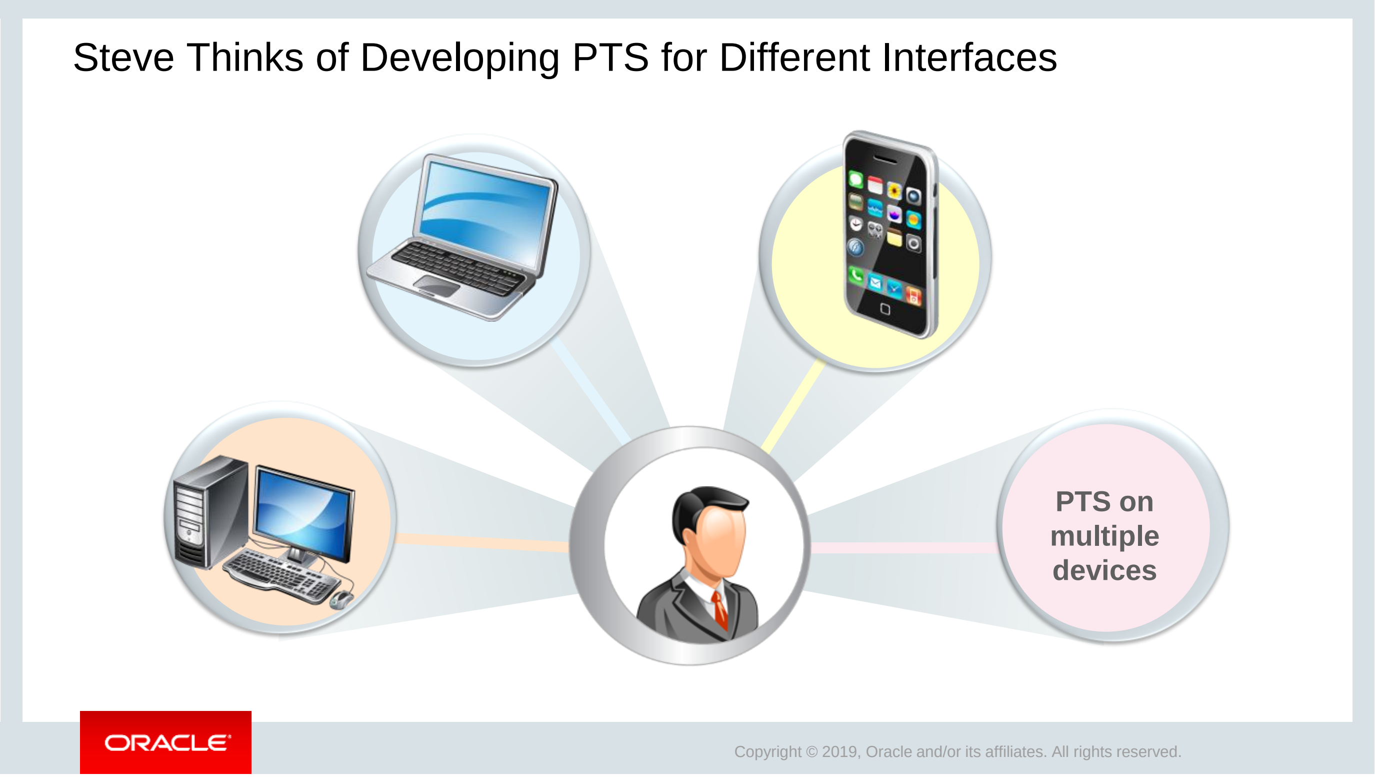
This slide shows a graphical representation of the entire course, highlighting lesson 3 in particular, which is dealt with in these slides.

Objectives

After completing this lesson, you should be able to:

- Identify the components of a database application
- Describe the database application user interfaces
- Explain the various ways of creating a database application
- Use App Builder to create a Database application

This lesson introduces you to **App Builder** and how to use it to create a database application. You learn about the different components of a database application along with the concepts associated with building a database application.



Steve Thinks of Developing PTS for Different Interfaces

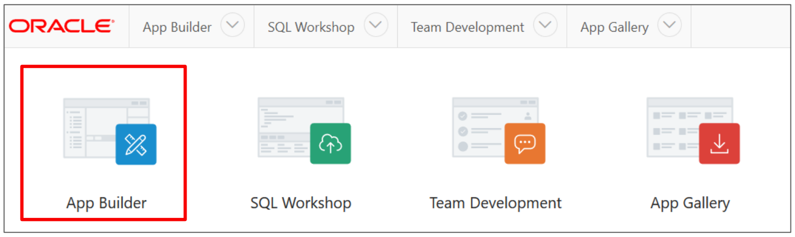
Now that Steve has a fairly good understanding of Oracle Application Express, he is quite confident that it is the correct tool for developing a database application, such as PTS. Steve also learns that by using Oracle Application Express, he can create an application with a responsive design that can be used both for database and mobile interface.

Lesson Agenda

- App Builder Overview
  - Accessing App Builder
  - App Builder Interface
- Introducing Database Applications
- Creating a Database Application

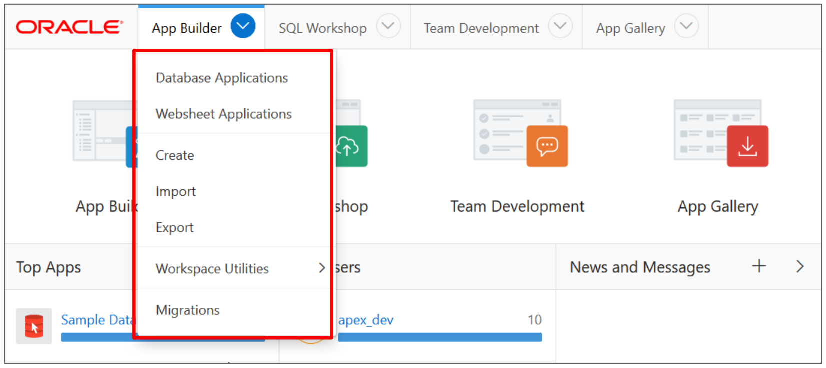
Accessing App Builder

a Quick Links



b

Drop-down list



Steve accesses the **App Builder**, a component of Application Express to create an application. He gets started with App Builder after signing in to the workspace.

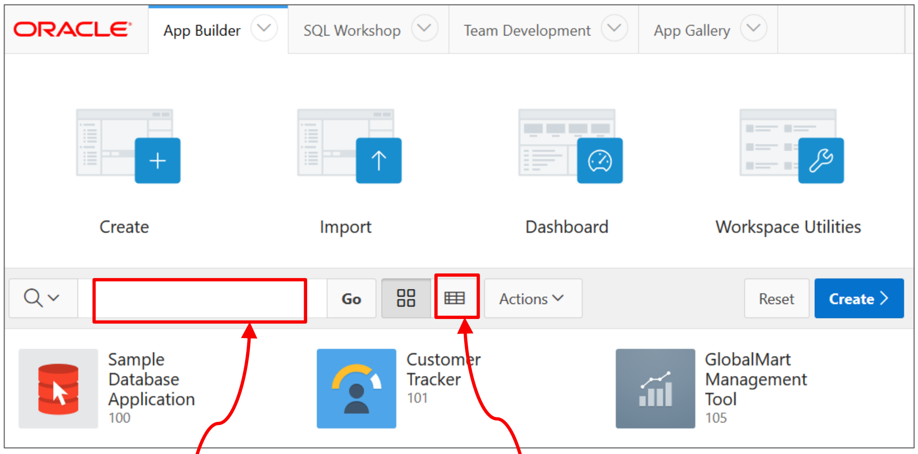
When signing in to Oracle Application Express, the Workspace home page opens. To access the **App Builder** home page, choose one of the following options:

- Click the **App Builder** icon to go to the App Builder home page (screenshot 1).
- Click the down arrow next to the App Builder tab to view the drop-down list. You can then select the appropriate menu option (screenshot 2).

App Builder Home Page

On the **App Builder** home page, you can:

- Create an application
- Import an application or files
- Access Workspace Utilities
- Search for an application
- Reset the application report
- Change the page view
- Use the Actions menu
- View or run an application

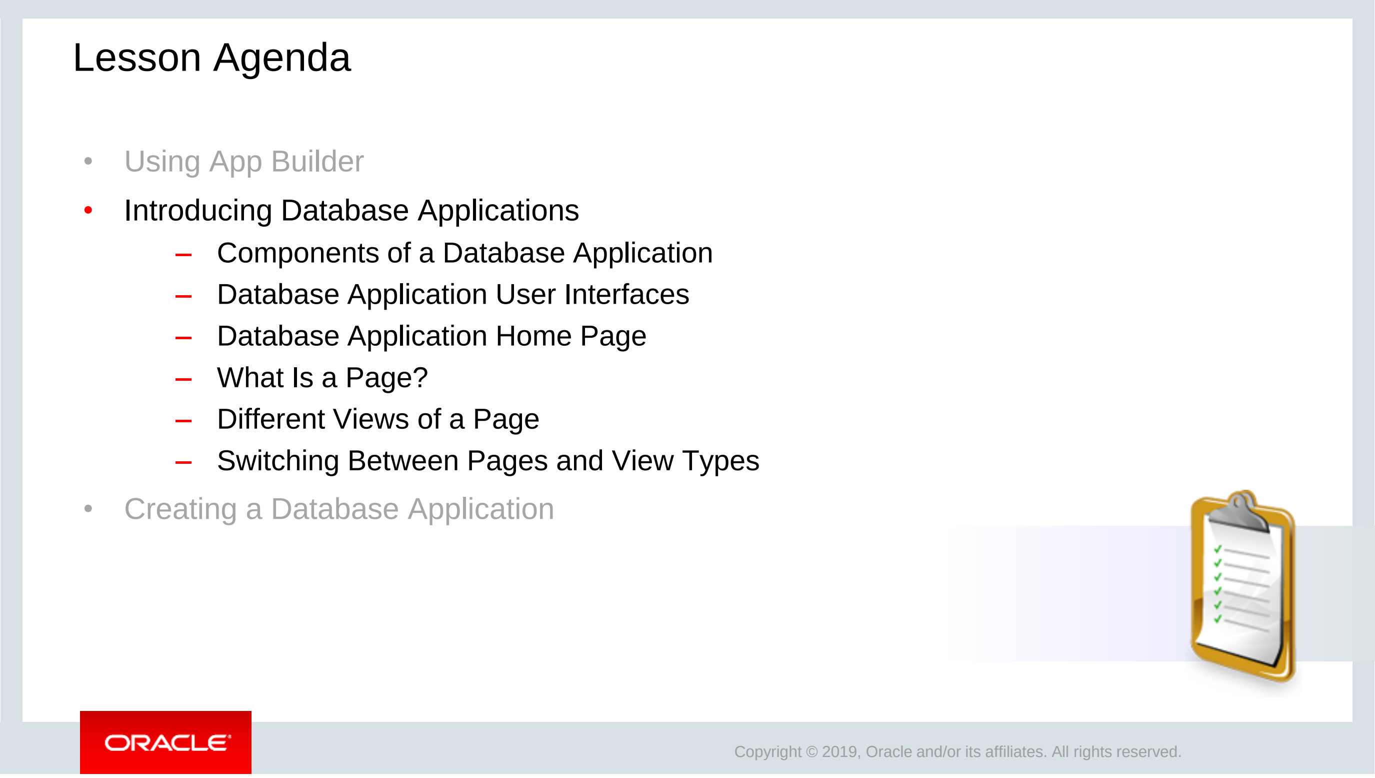


Search area

Page view

The **App Builder** home page displays the currently installed applications. On the App Builder home page, you can:

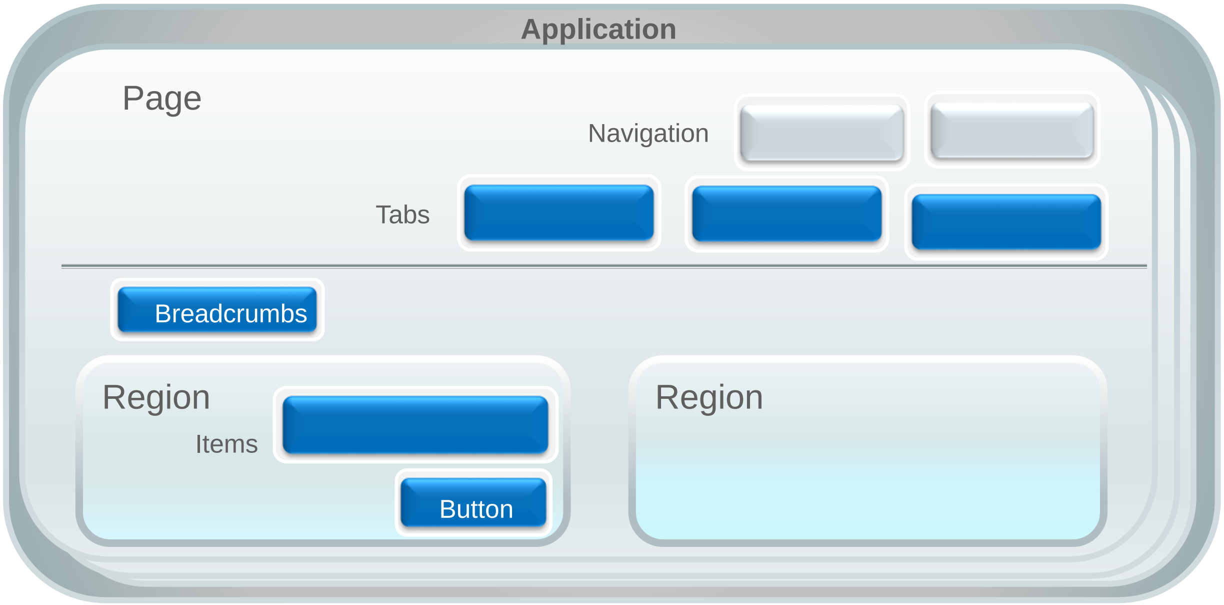
- **Search for an application:** Enables you to search for a particular application. Enter the name of the application in the Search area and click Go. You can also search a particular column by clicking the magnifying glass icon and selecting a column to search on. If no column is selected, then all the columns are searched.
- **Change the page view:** To change the appearance of a page by making a selection from the two View icons next to the Go button. These icons consist of:
  - **View icons** (the default)**:** Display each application as an icon and identify it by the application name
  - **View Report:** Displays a list of applications in a report
- **Use the Actions menu:** Enables you to perform different tasks for the data that is displayed
- **Reset the application report:** Returns you to the default display
- **Import an application:** Imports an exported application file. This enables you to import applications, Plug-Ins, Themes, User Interface Defaults, and Team Development Feedback into the workspace.
- **Create an application:** Creates a new application or installs a sample application
- **View or run an application:** Enables you to view a specific application. This opens the home page of that application. Selecting any application on the App Builder home page highlights a small Run icon, by clicking which you can run the application without even opening it.



Lesson Agenda

- Using App Builder
- Introducing Database Applications
  - Components of a Database Application
  - Database Application User Interfaces
  - Database Application Home Page
  - What Is a Page?
  - Different Views of a Page
  - Switching Between Pages and View Types
- Creating a Database Application

Components of a Database Application



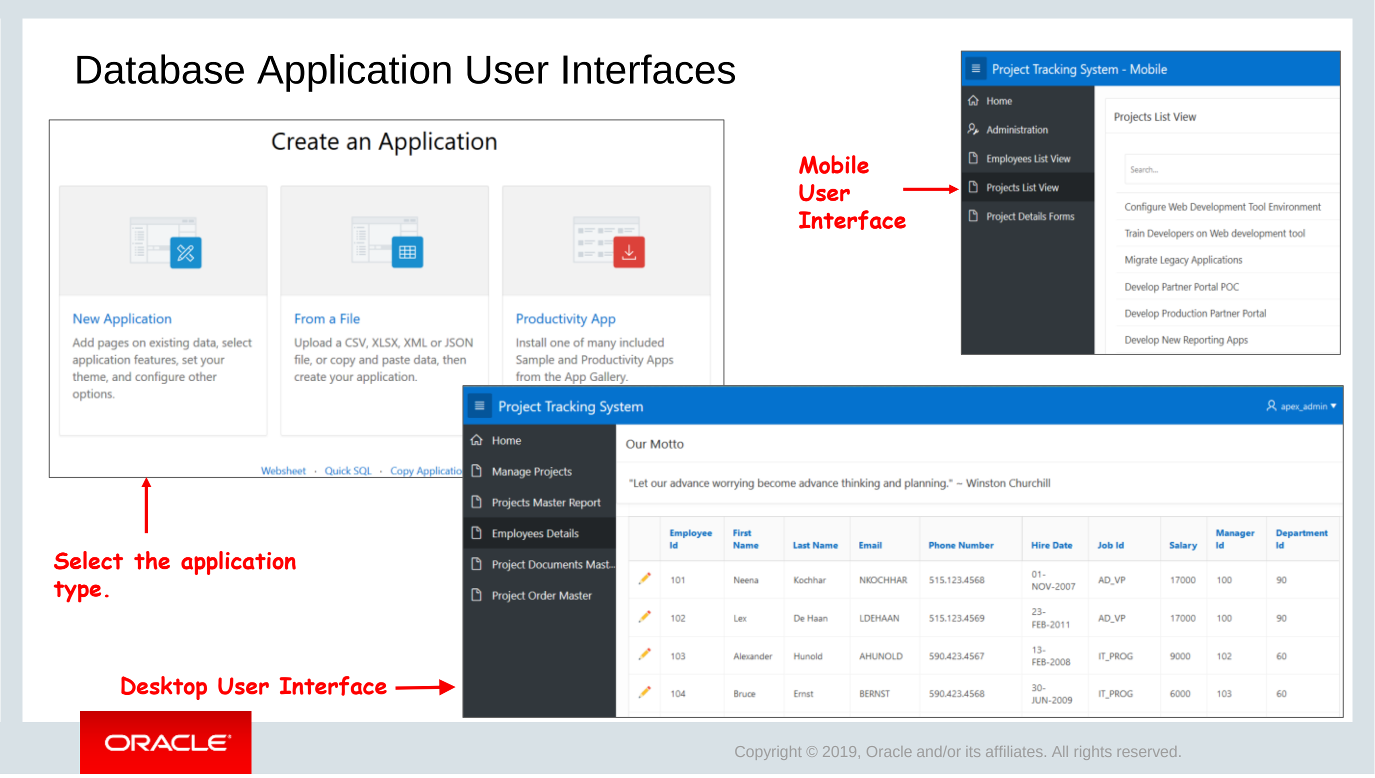
A database application is a collection of database-driven webpages that are linked by navigational controls, such as tabs, buttons, and hypertext links.

A **page** is the basic element of an application. A page is divided into regions; a **region** is a section of a page that contains content. The content of the region is determined by the region source. For example, a region can contain a report based on a SQL query, or it can contain static HTML.

A region can also contain the following:

- **Items** such as a text field, text area, select list, and check box
- **Buttons** to direct users to a specific page or URL and also to post and process information
- **Breadcrumbs** (locator links) to provide hierarchical navigation

Navigation entries are placed outside regions to enable users to navigate between the pages of an application.



Database Application User Interfaces

Create an Application

New Application
Add pages on existing data, select application features, set your theme, and configure other options.

From a File
Upload a CSV, XLSX, XML or JSON file, or copy and paste data, then create your application.

Productivity App
Install one of many included Sample and Productivity Apps from the App Gallery.

Select the application type.

Mobile User Interface

Project Tracking System - Mobile

Projects List View

- Configure Web Development Tool Environment
- Train Developers on Web development tool
- Migrate Legacy Applications
- Develop Partner Portal POC
- Develop Production Partner Portal
- Develop New Reporting Apps

Desktop User Interface

Project Tracking System

Our Motto

"Let our advance worrying become advance thinking and planning." ~ Winston Churchill

	Employee Id	First Name	Last Name	Email	Phone Number	Hire Date	Job Id	Salary	Manager Id	Department Id
✎	101	Neena	Kochhar	NKOCHHAR	515.123.4568	01-NOV-2007	AD_VP	17000	100	90
✎	102	Lex	De Haan	LDEHAAN	515.123.4569	23-FEB-2011	AD_VP	17000	100	90
✎	103	Alexander	Hunold	AHUNOLD	590.423.4567	13-FEB-2008	IT_PROG	9000	102	60
✎	104	Bruce	Ernst	BERNST	590.423.4568	30-JUN-2009	IT_PROG	6000	103	60

At the time of creating a new database application, you must click **New Application.**

If you plan to create an application from a spreadsheet, then click **From a Spreadsheet**. If you want to get started with a Productivity application, then click the third icon called Productivity App. In this lesson, we focus on creating a database application.

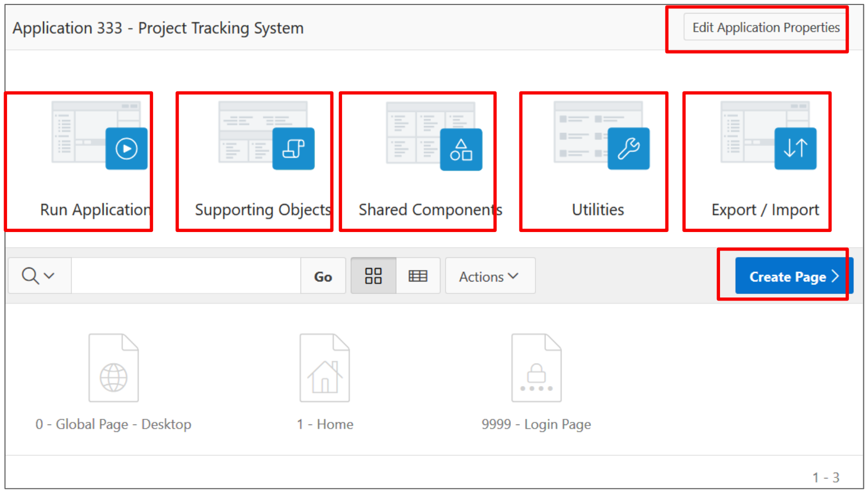
Websheet applications are different from database applications. They allow users to create data-centric applications without any SQL programming knowledge.

Before you get started with creating a database application, let's take a quick tour of some of the Application Express features and interfaces that you would be using to create the application. These are Themes, Page Designer, and database components such as pages, tabs, regions, items, buttons, navigation, and so on.

Database Application Home Page

On an application home page, you can:

- Run the application
- Use the Supporting Objects utility
- Create shared components
- Create a page
- Examine application utilities
- Export and import applications
- Edit application properties



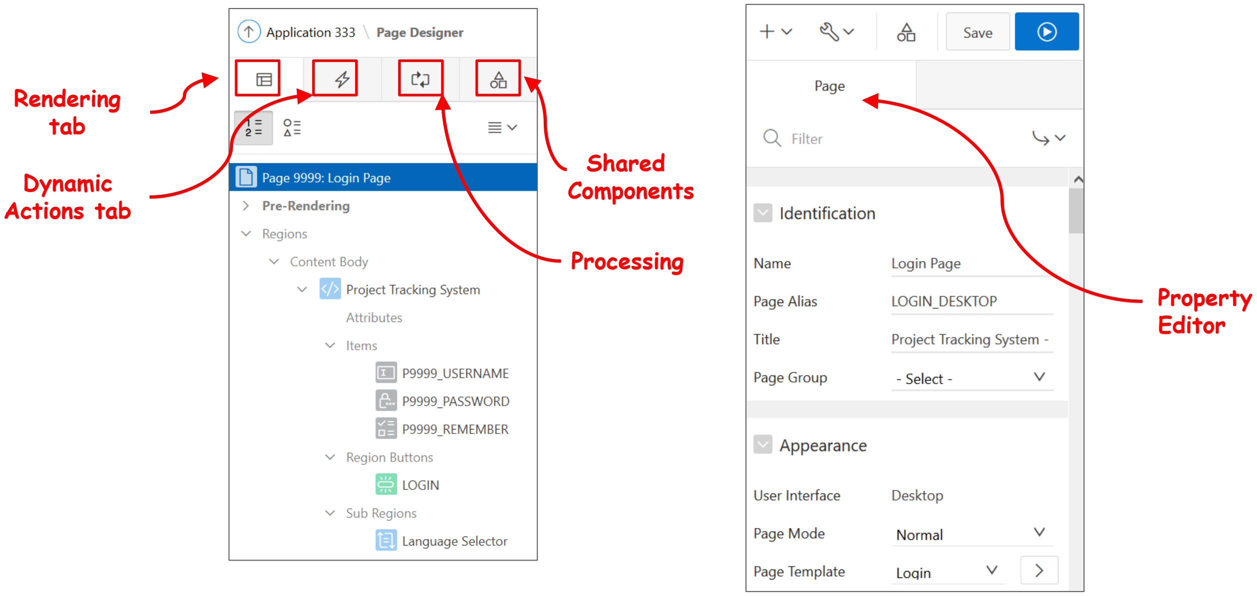
This slide shows the application home page of the *Project Tracking System* application that Steve created. On the App Builder home page, when you click the application icon or application name, the home page of that particular application opens. You can see the application ID and the name of the application at the top of the page, as highlighted in the slide.

From the application home page, you can:

- **Run the application:** Click the **Run Application** icon to submit the pages to the Oracle Application Express engine to render a viewable HTML page.
- **Use the Supporting Objects utility:** Click **Supporting Objects** to access the utility to define the database object definitions, images, and seed data to be included in your application export for your packaged application.
- **Create shared components:** Click **Shared Components** to build shared application components and user interface controls.
- **Examine application utilities:** Click **Utilities** to monitor developer activity, view dashboards, run the Advisor, and view numerous other reports.
- **Export and import an application:** Click the **Export/Import** icon to export or import an entire application or its components, such as cascading style sheets, images, static files, themes, and user interface defaults.
- **Edit application properties:** Click **Edit Application Properties** to edit the application name and availability and to define static substitution strings. Additionally, the Edit Application page displays the defined build options, the associated theme, template defaults, and component defaults.
- **Create a page:** Click **Create Page** to add a page to your application.

On the application home page, you also see a list of icons for each page. To open a page, click the corresponding page icon.

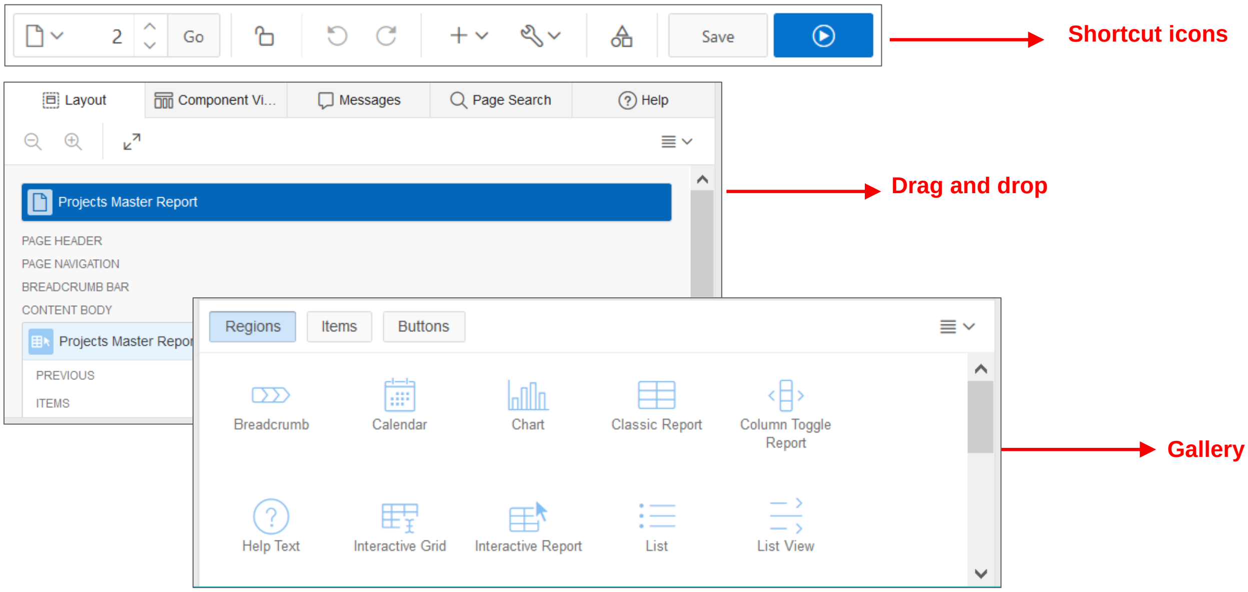
Page Designer: The World’s Most Advanced IDE



Name	Login Page
Page Alias	LOGIN_DESKTOP
Title	Project Tracking System -
Page Group	- Select -
User Interface	Desktop
Page Mode	Normal
Page Template	Login

Page items, regions, computations, processes, and **validations** are all organized under **Rendering** and **Processing** tabs automatically. **Dynamic actions** and **Shared Components** are organized under separate tabs. Select a component in **Component Selector** and update its properties in **Property Editor.**

Page Designer: Drag and Drop from Gallery

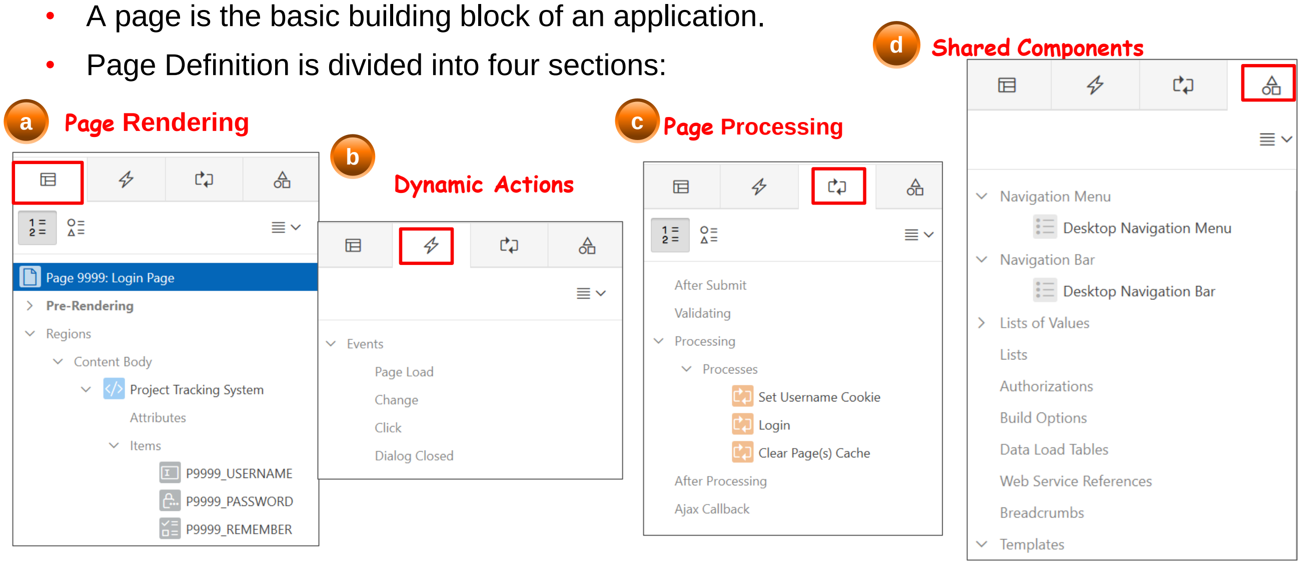


Page Designer provides many useful features to developers:

- You can **drag and drop** regions, items, and buttons into **grid layout**. All the related **properties** will be updated automatically.
- It supports backward compatibility with the **Component view**. You can easily view the page in the component view just by clicking one icon.
- The **Help tab** provides help on any attribute in the property editor of a selected component.
- **Shortcut icons** enable you to create a new page, copy page, create any type of region on the current page, shared component, bug, feedback, and so on. They also enable you to access debugging tools and other utilities.
- **Undo** enables you to undo an update, and **Redo** reapplies the last update that was undone using **Undo**.
- **Errors and Warnings** are immediately visible.

Page Definition: Overview

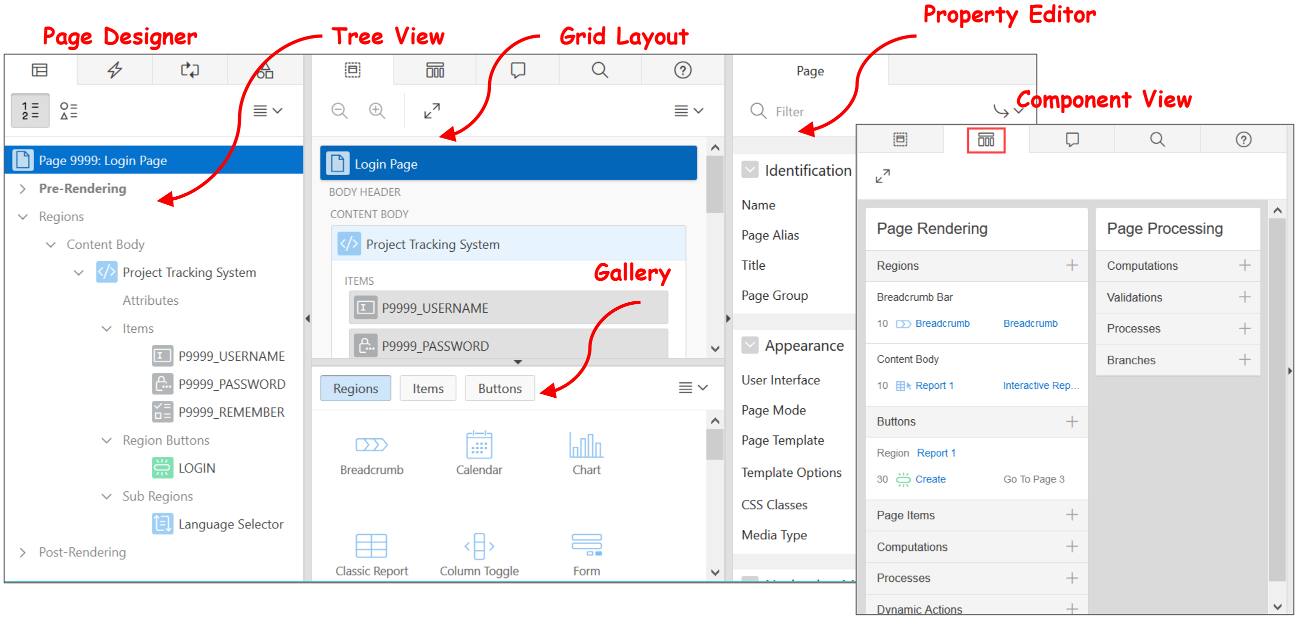
- A page is the basic building block of an application.
- Page Definition is divided into four sections:



You build an application by using pages. Page Definition is divided into:

- **Page Rendering:** The process of generating a page from the database. You can use the Page Rendering section to modify the controls that impact the rendering of a page, including page definition, regions, buttons, items, page-rendering computations, and page processes.
- **Page Processing:** The process of submitting a page. A page is submitted when a user clicks a button. You can use the Page Processing section of Page Definition to specify application logic, such as computations, validations, processes, and branches. In general, the Application Express engine runs the logic of specific applications in the order in which they appear on the Page Definition.
- **Shared Components:** List of the common components that can be displayed or applied on every page within an application. Some of the shared components include tabs, lists of values, breadcrumbs, lists, and navigation bars.

Different Views of a Page



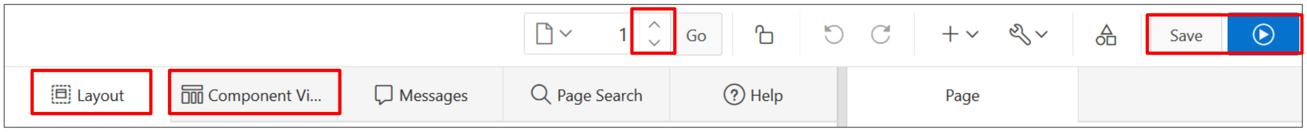
There are two ways to view a page: Page Designer and Component view.

- The **Page Designer** organizes the page components under four major categories:
    - **Tree View:** The components can be seen in tree view organized under Page Rendering, Dynamic Actions, Page Processing, and Shared Components.
    - **Grid Layout:** It gives a snapshot of how the page is going to look as per the design. You can drag an item or region or button to the required position on the grid layout. It also has tabs to show error messages (if any), Page Search, and Help on an attribute selected in the property editor for any item.
    - **Gallery:** The gallery contains all possible types of Regions, Items, and Buttons, which can be designed on this page using grid layout.
    - **Property Editor:** This is located on the left side of the Page Designer and shows a complete list of all the properties of a component selected in the Page Components on the right side of the page.
- The **Component view** groups user interface elements and application logic by component type.

Switching Between Pages and View Types

The navigation bar enables you to:

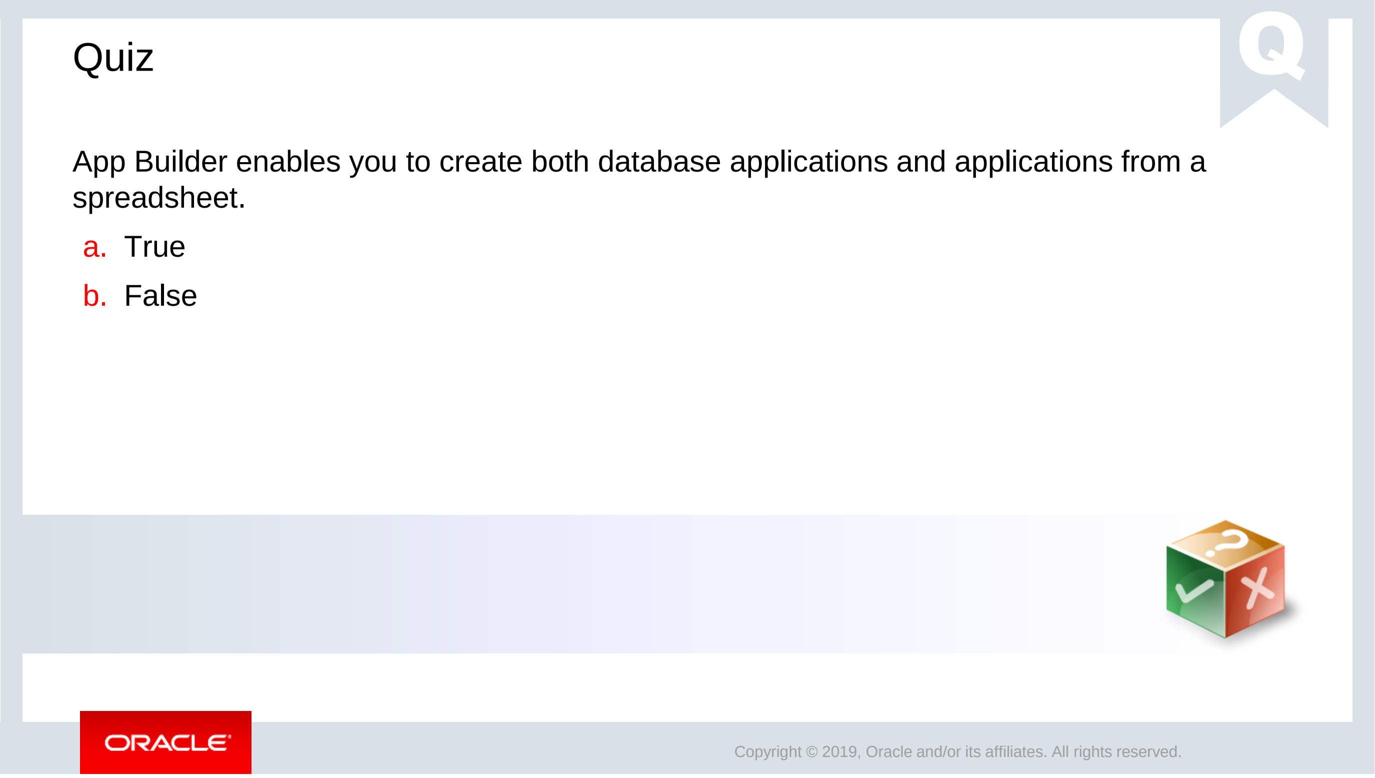
- Specify a specific page
- Undo or Redo a recent change
- Change view types
- Use shortcut links to create another Page, Region, Items, and so on
- Access debugging tools
- Lock or unlock the page
- **Save and Run Page**



The following are the three ways to switch from one page to another:

1. Enter a page number in the **Page** field and then click **Go.**
2. Click the up arrow next to the **Page** field and then select a page from the list.
3. Click the **Previous** and **Next** buttons to the right of the Go button.

To switch from the Layout view (default) to the Component view, click **Component View** on the navigation bar. To switch to the Layout view, click **Layout** on the navigation bar.



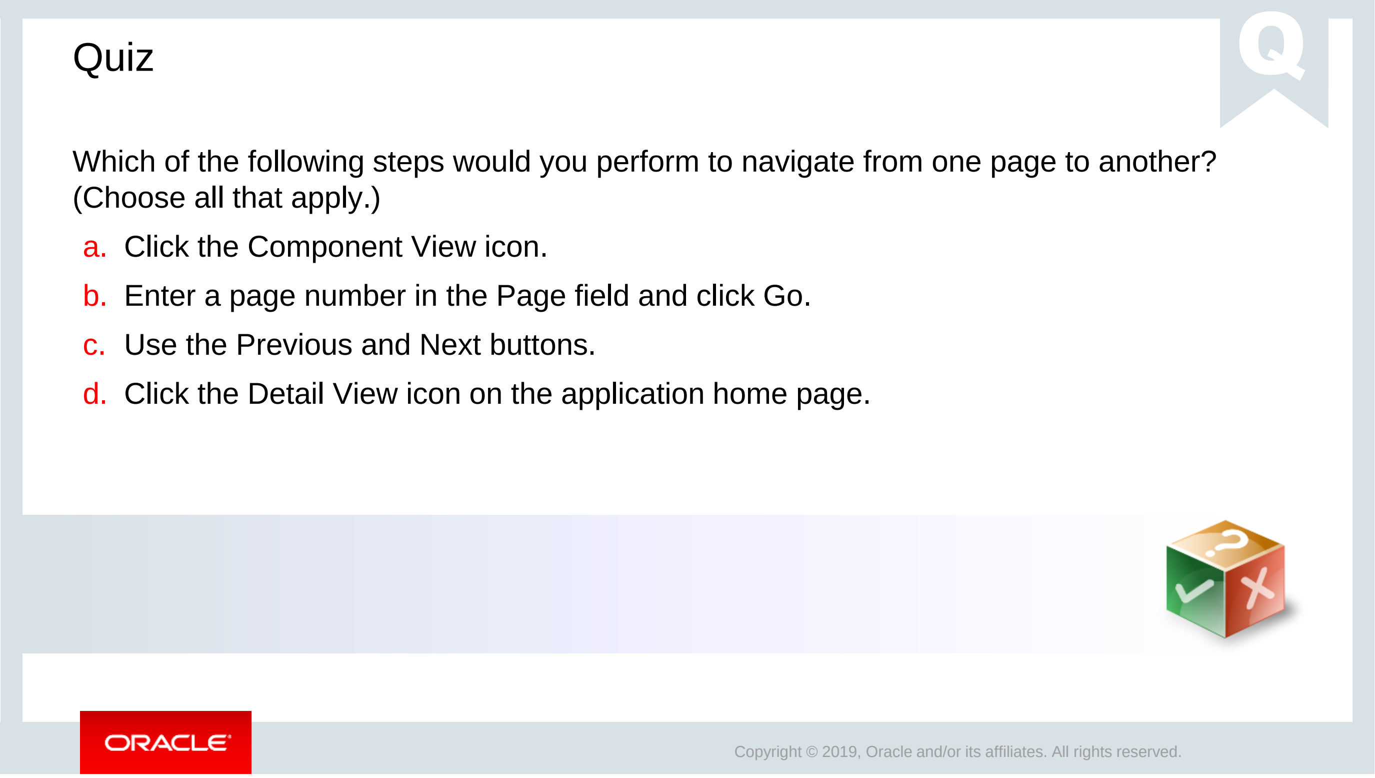
Quiz

App Builder enables you to create both database applications and applications from a spreadsheet.

a. True

b. False

Answer: a



Quiz

Which of the following steps would you perform to navigate from one page to another? (Choose all that apply.)

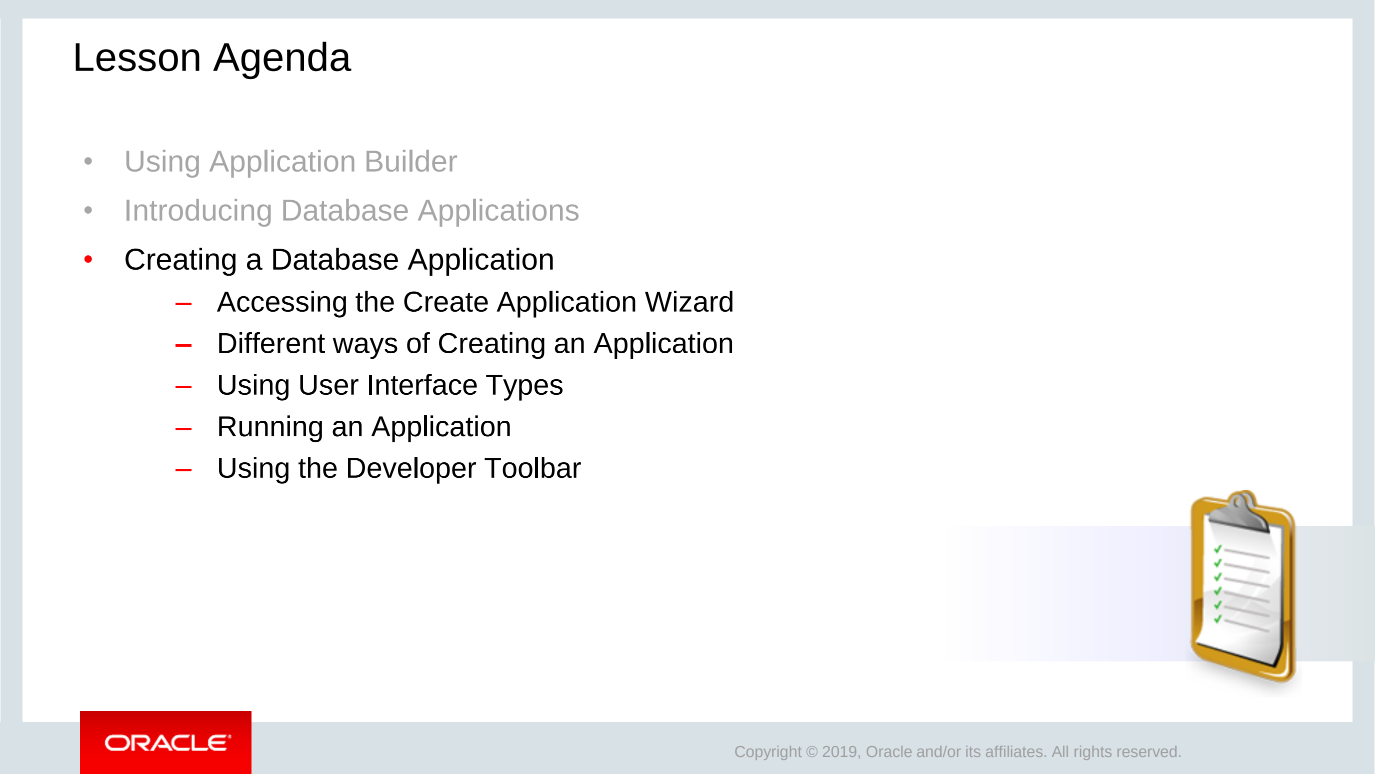
a. Click the Component View icon.

b. Enter a page number in the Page field and click Go.

c. Use the Previous and Next buttons.

d. Click the Detail View icon on the application home page.

Answer: b, c

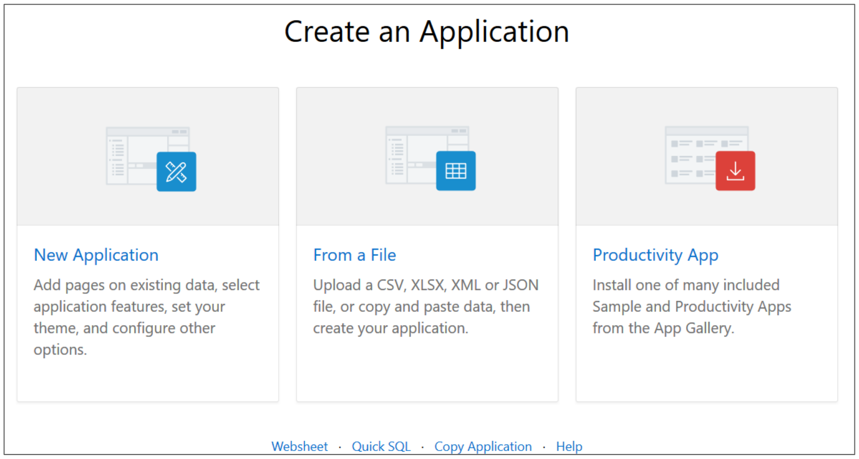


Lesson Agenda

- Using Application Builder
- Introducing Database Applications
- Creating a Database Application
  - Accessing the Create Application Wizard
  - Different ways of Creating an Application
  - Using User Interface Types
  - Running an Application
  - Using the Developer Toolbar

Create Application Wizard

- The Create Application Wizard allows you to:
  - Create a new database application
  - Create an application from a file
  - Create a productivity app
  - Create an application from a websheet
  - Create an application using quick SQL
  - Create a copy of an application



Create an Application

New Application

Add pages on existing data, select application features, set your theme, and configure other options.

From a File

Upload a CSV, XLSX, XML or JSON file, or copy and paste data, then create your application.

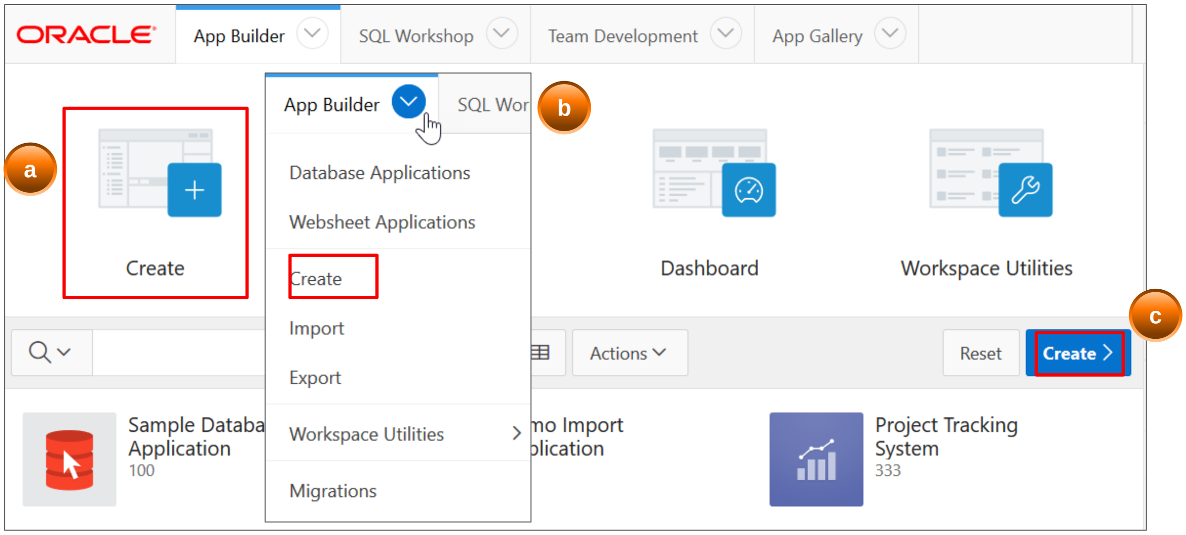
Productivity App

Install one of many included Sample and Productivity Apps from the App Gallery.

You create an Oracle Application Express application by running the **Create App Wizard.** To run the Create App Wizard:

1. Click **Create** in the App Builder home page. This opens the Create App Wizard.
2. In the next step in the wizard, select the type of application that you want to create:
   - **New application** – To create a database application and mobile application. This is the option you select in this lesson to create the database application.
   - **From a File** – To load spreadsheet data from a file as comma separated (*.csv) or tab delimited file and then create an interactive report on that data
   - **Productivity App** – To install and run packaged applications that includes sample applications, productivity applications, and custom applications
   - **Websheet** – To create an application from a websheet
   - **Quick SQL** – To create an application using SQL. Sample SQL codes are also available.
   - **Copy Application** – To make a copy of an existing application

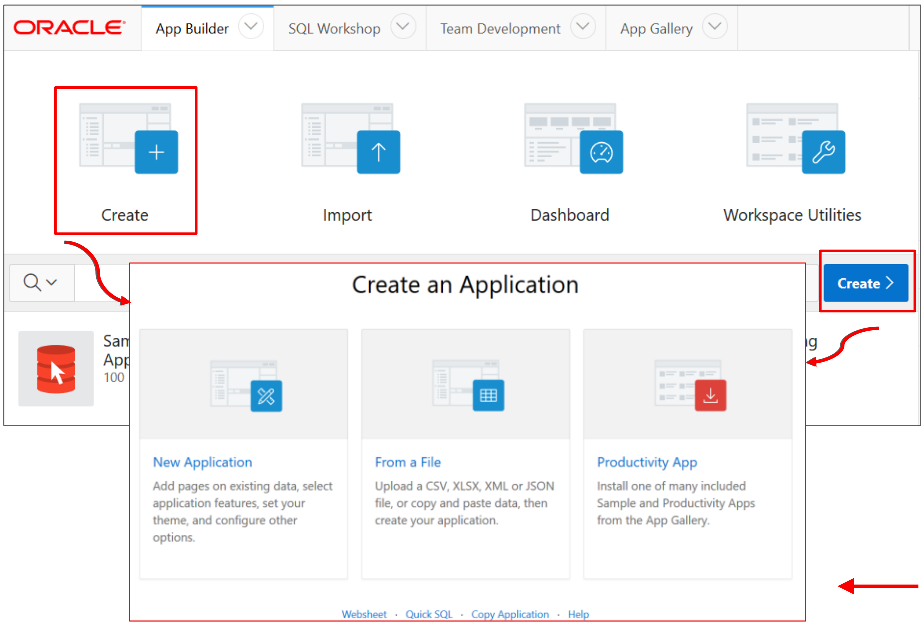
Accessing the Create Application Wizard



You can access the Create App Wizard by:

- Navigating to the App Builder home page and clicking **Create** icon (screenshot a).
- Selecting **Create** from the App Builder drop-down list (screenshot b).
- Navigating to the App Builder home page and clicking the **Create** button (screenshot c).

Different Ways of Creating a Database Application



New Application
Add pages on existing data, select application features, set your theme, and configure other options.

From a File
Upload a CSV, XLSX, XML or JSON file, or copy and paste data, then create your application.

Productivity App
Install one of many included Sample and Productivity Apps from the App Gallery.

App Builder home page

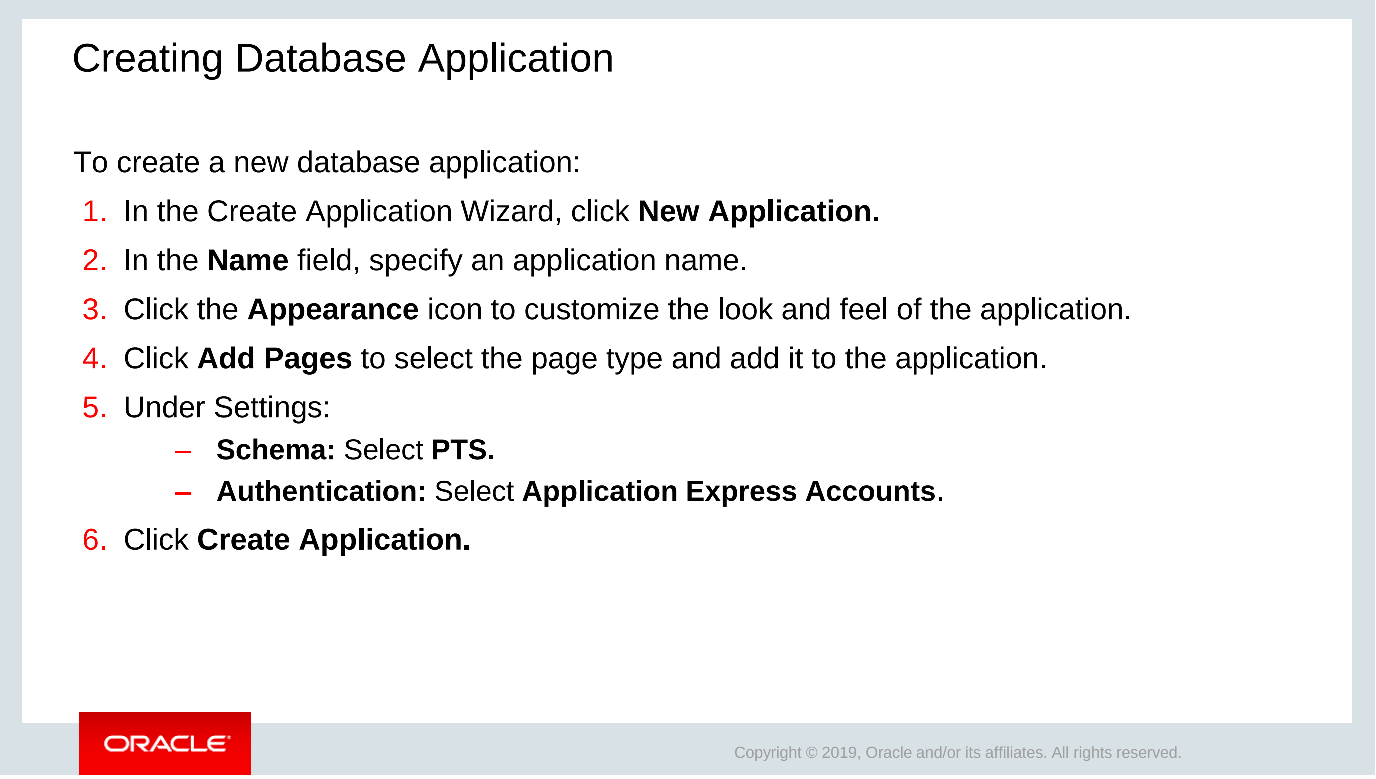
Create App Wizard

To create a database application, select Database for the application type and then click Next. You have four options to create a database application:

- **From a File:** You can create an application based on data from a `.csv, .xlxs, .xml,` or `.json` file. You first upload or paste the data to create a table. Then you select a default appearance. The resulting application enables end users to query, insert, or update records or analyze the data.
- **Copy of existing Application:** You can create a copy of an existing application by running the Create Application Wizard or by selecting the application and then copying the application on the Application home page.
- **Install sample applications:** Oracle Application Express includes several sample applications. You can install, run, and use sample applications as they are or analyze them to better understand how to use Application Builder to build specific types of functionality.

You can also create a database application based on a table, query, or a drill-down query.

Because Steve decides to develop a project management tool for his manager Stella, he creates a new application in this step and names it *Project Tracking System*



Creating Database Application

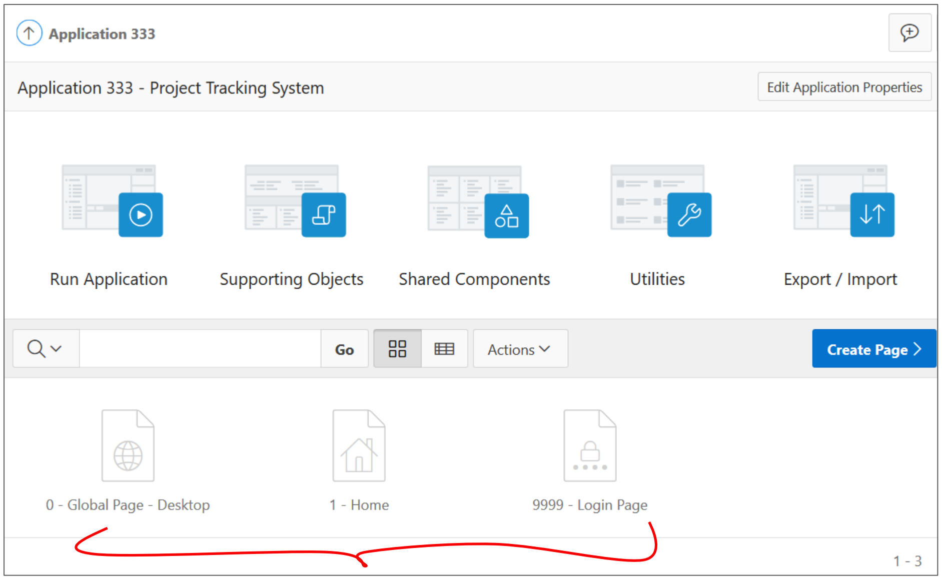
To create a new database application:

1. In the Create Application Wizard, click **New Application.**
2. In the **Name** field, specify an application name.
3. Click the **Appearance** icon to customize the look and feel of the application.
4. Click **Add Pages** to select the page type and add it to the application.
5. Under Settings:
   - **Schema:** Select **PTS.**
   - **Authentication:** Select **Application Express Accounts**.
6. Click **Create Application.**

When running the Create Application Wizard, you must choose a target user interface based on which the wizard is optimized to display the appropriate page types, attributes, and themes. The slide provides an overview of the steps to create a database application based on a table, query, or a drill-down query.

Steve successfully creates a new database application called PROJECT TRACKING SYSTEM (PTS) by using App Builder.

Project Tracking System Home Page



Default pages of the PTS application

The slide shows the page wizard for a database application. The page wizard displays the various page types available based on the type of user interface.

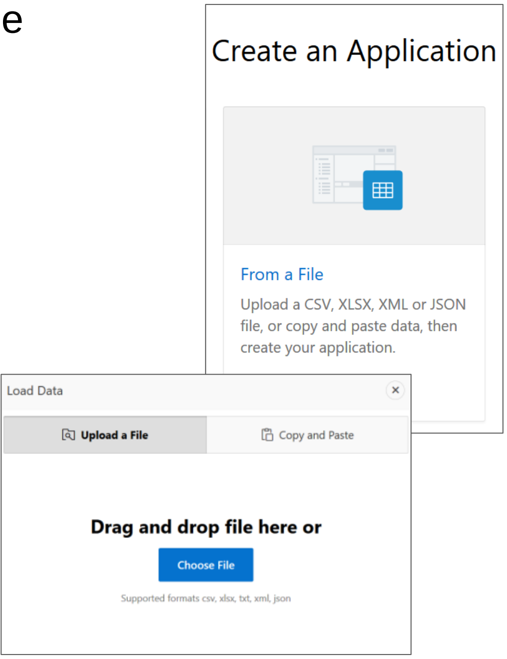
At the time of creation, the PTS application has three pages created by default:

- **0 - Global Page – Desktop**
- **1- Home:** The Home page acts as the parent of any new pages added to the application. The page number for the Home page is always 1.
- **9999 - Login page:** The login page gets created at the time of creating an application. This page is used to enter the login credentials of the application. The page number for the login page is 9999.

Creating a Database Application from a File

In the Create Application Wizard, after clicking **From a File**, perform the following steps:

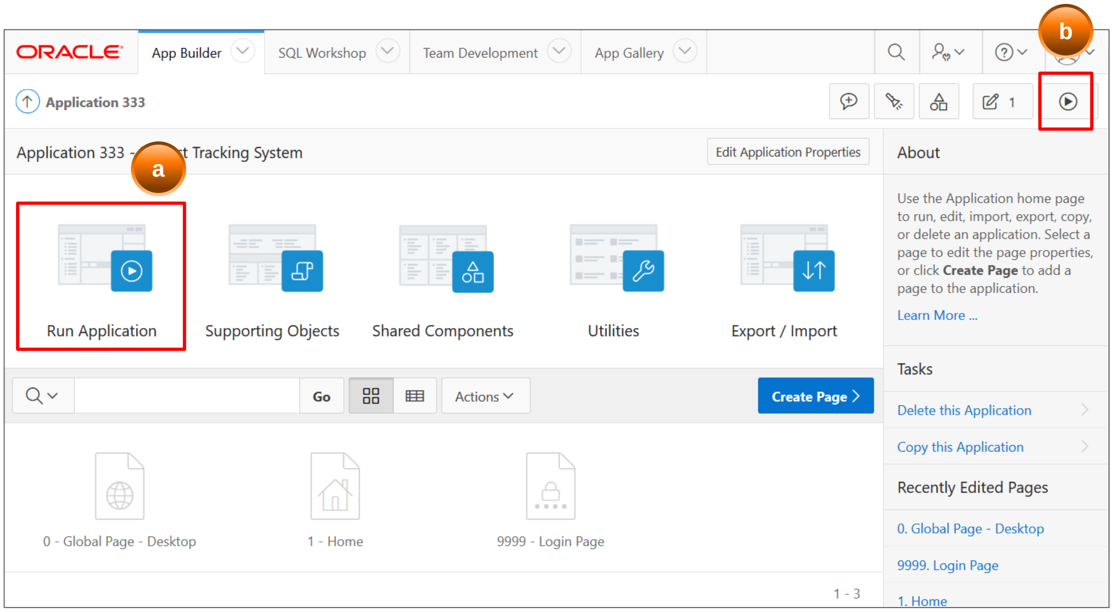
1. Specify how the data will be loaded.
   - Upload a file or
   - Copy and paste the data
2. Specify the table name and column specifications.
3. Specify user interface defaults.
4. Enter the application name.
5. Select a theme.
6. Specify whether you want the data to be summarized, as well as which columns to use.



Upload a CSV, XLSX, XML or JSON file, or copy and paste data, then create your application.

In Oracle Application Express 19.1, you can also create an application from a file such as a `.CSV`, `.XLSX`, `.XML`, or `.JSON` file by clicking **From a File** in the Create App Wizard. You can also use this option to copy paste your data to create the application. The slide provides an overview of the steps that are necessary to create a database application from a spreadsheet.

Running an Application



App Builder

SQL Workshop

Team Development

App Gallery

Application 333

Application 333 – [a] t Tracking System

Edit Application Properties

About

Use the Application home page to run, edit, import, export, copy, or delete an application. Select a page to edit the page properties, or click **Create Page** to add a page to the application.

Learn More ...

Tasks

Delete this Application

Copy this Application

Recently Edited Pages

0. Global Page - Desktop

9999. Login Page

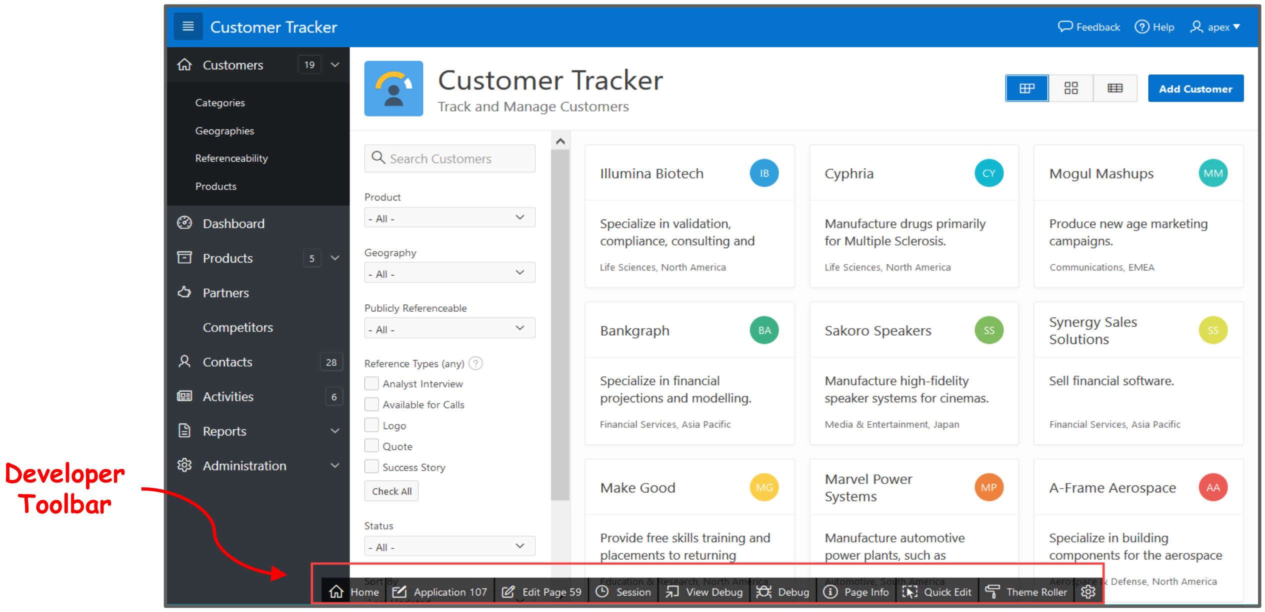
1. Home

Regardless of the application that you create, you can run the application in two ways:

- By clicking the **Run Application** icon with *Run Application* written on it
- By clicking the **Run** icon at the top-right side of the Application Page

Note: If you have chosen the Application Express authentication scheme, the Sign In page appears. Enter your workspace username and password and click **Sign In** to sign in to your application.

Using the Developer Toolbar



Illumina Biotech IB
Specialize in validation, compliance, consulting and
Life Sciences, North America

Cyphria CY
Manufacture drugs primarily for Multiple Sclerosis.
Life Sciences, North America

Mogul Mashups MM
Produce new age marketing campaigns.
Communications, EMEA

Bankgraph BA
Specialize in financial projections and modelling.
Financial Services, Asia Pacific

Sakoro Speakers SS
Manufacture high-fidelity speaker systems for cinemas.
Media & Entertainment, Japan

Synergy Sales Solutions SS
Sell financial software.
Financial Services, Asia Pacific

Make Good MG
Provide free skills training and placements to returning

Marvel Power Systems MP
Manufacture automotive power plants, such as

A-Frame Aerospace AA
Specialize in building components for the aerospace
Defense, North America

Developer Toolbar

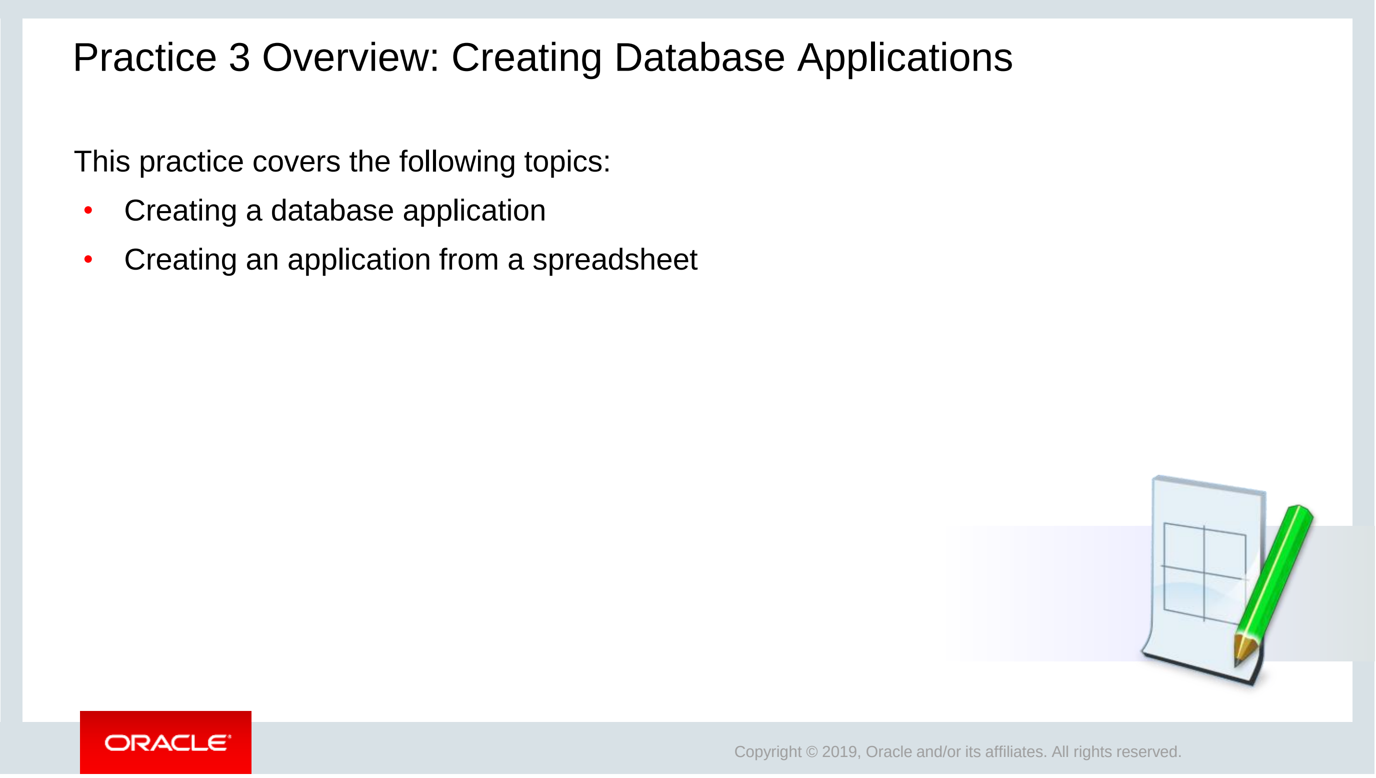
Developers who run an application directly from App Builder have access to the Developer toolbar. Any users running the application directly, using the runtime URL, will not see the Developer toolbar. The Developer toolbar offers a quick way to accomplish the following:

- Edit the currently running page
- Create a new page, control, or component
- View session state
- Toggle the edit links on and off

The page displayed in this slide is the home page of the Customer Tracker Packaged Application in Oracle Application Express. The Developer toolbar is displayed at the bottom of every page in a running application and has the following options:

- **Home:** Opens the Workspace home page
- **Application <n>:** Opens the application home page
- **Edit Page <n>:** Accesses the Page Definition for the current page
- **Session:** Displays a new window that contains session state information for the current page
- **View Debug:** Displays another window with debug information by session
- **Debug:** Toggles the page between Debug and No Debug mode. To view the debug information after Debug is selected, click View Debug.
- **Quick Edit:** Allows you to choose any component on the current page and opens the page with that component selected for editing
- **Theme Roller:** Provides options to change the look and feel of the user interface of the application and allows you to save your private themes created using theme roller

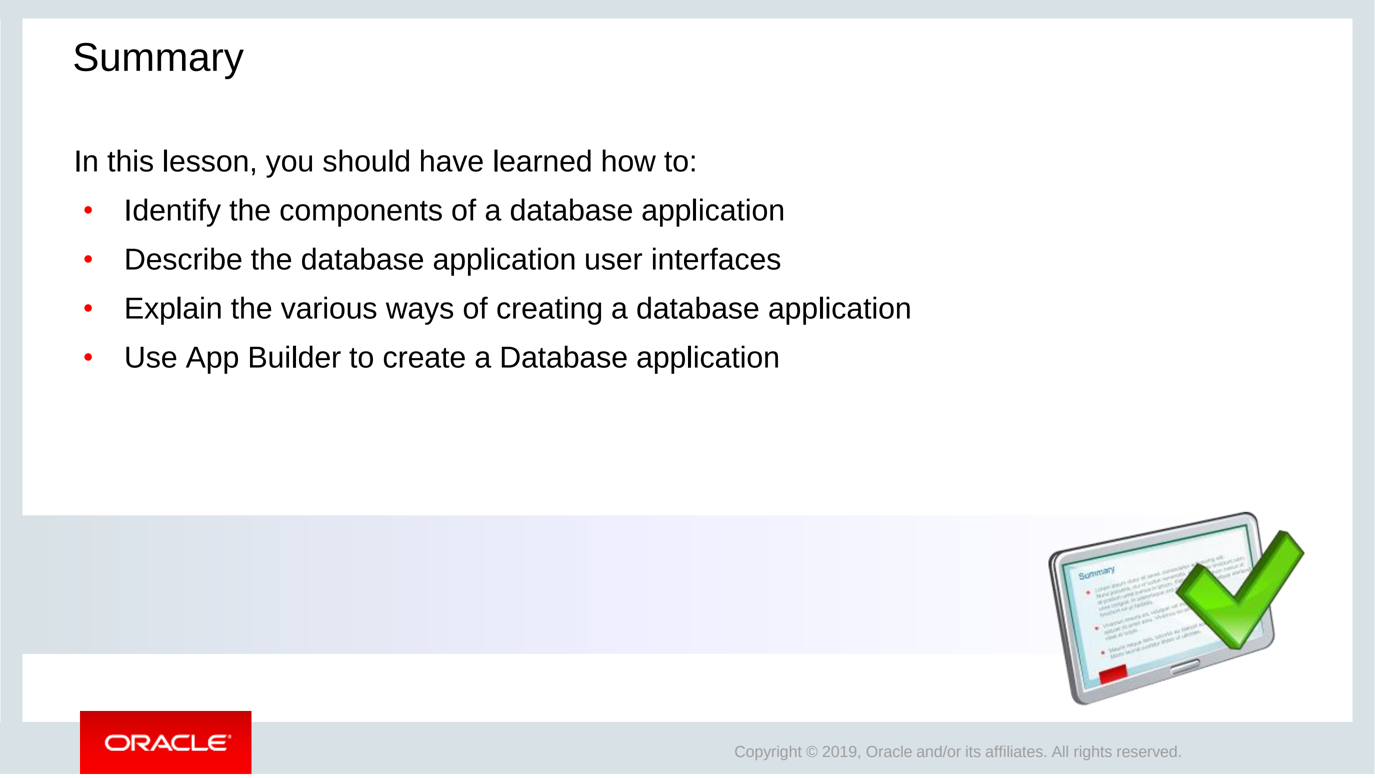
  Note: Theme Roller is discussed in more detail in the lesson titled “Working with Themes, Templates, and Files” in Unit 3 of this course.



Practice 3 Overview: Creating Database Applications

This practice covers the following topics:

- Creating a database application
- Creating an application from a spreadsheet



Summary

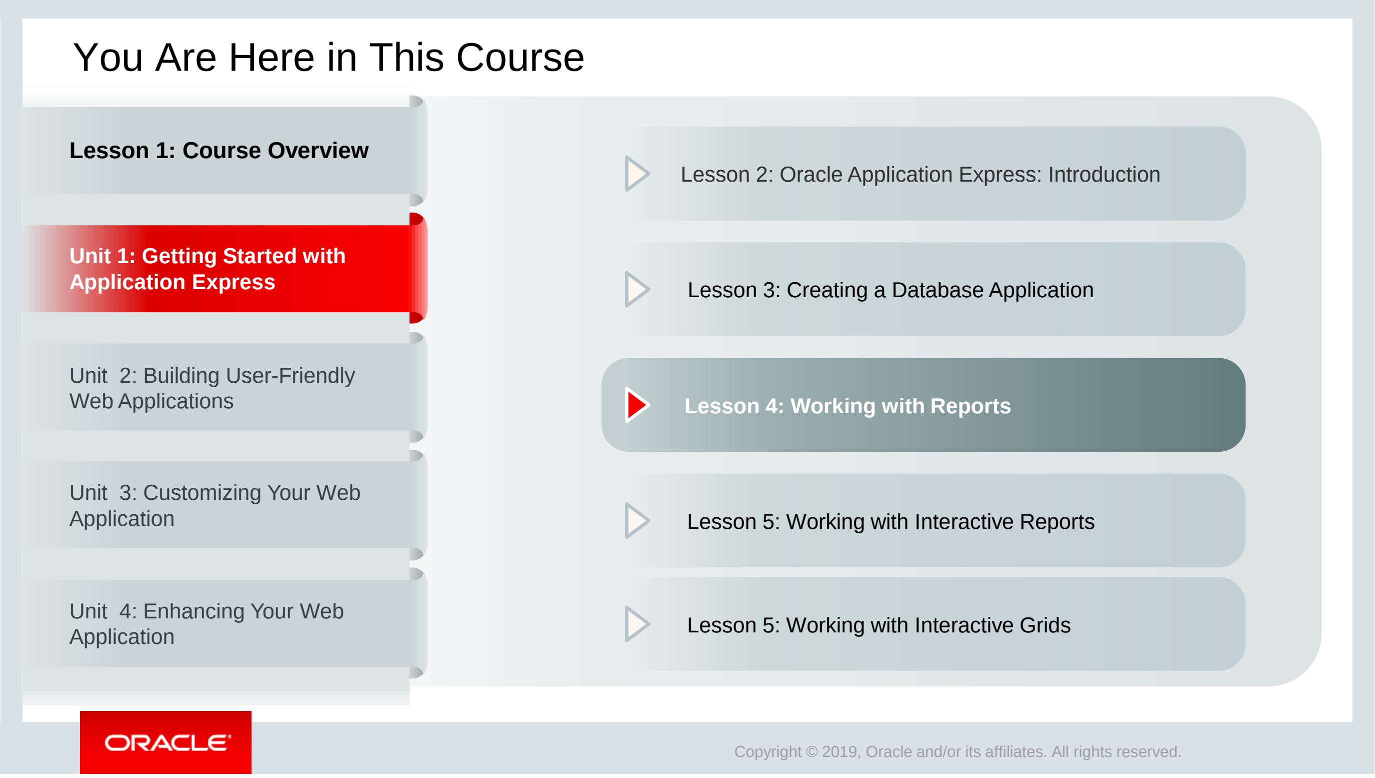
In this lesson, you should have learned how to:

- Identify the components of a database application
- Describe the database application user interfaces
- Explain the various ways of creating a database application
- Use App Builder to create a Database application

This lesson introduced you to App Builder. You learned about the different types of applications that you can build and the various components of an application. You also learned how to create a database application by using the Create Application Wizard from App Builder.

4

Working with Reports



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

Lesson 2: Oracle Application Express: Introduction

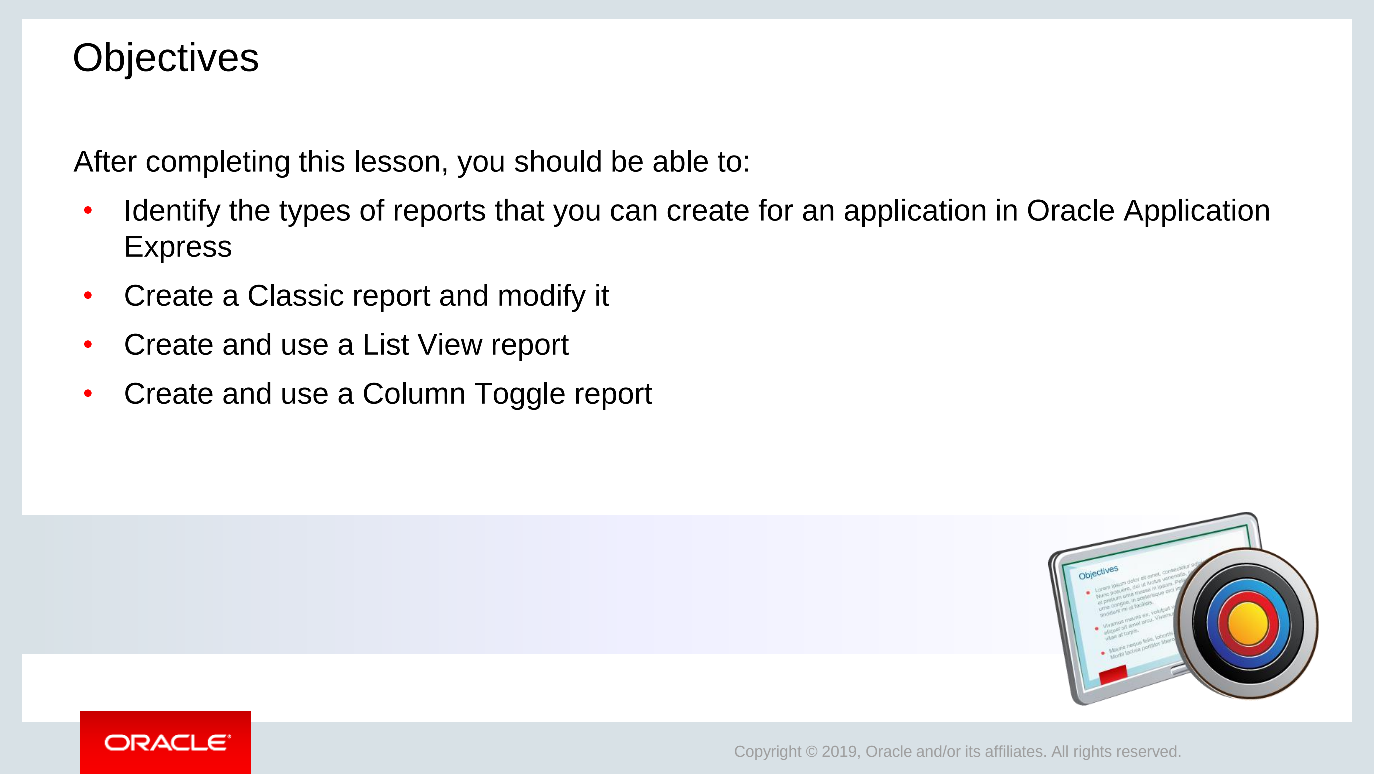
Lesson 3: Creating a Database Application

Lesson 4: Working with Reports

Lesson 5: Working with Interactive Reports

Lesson 5: Working with Interactive Grids

This slide is a graphical depiction of the course, highlighting Unit 1 - Lesson 4 in particular, which is dealt with in these slides.

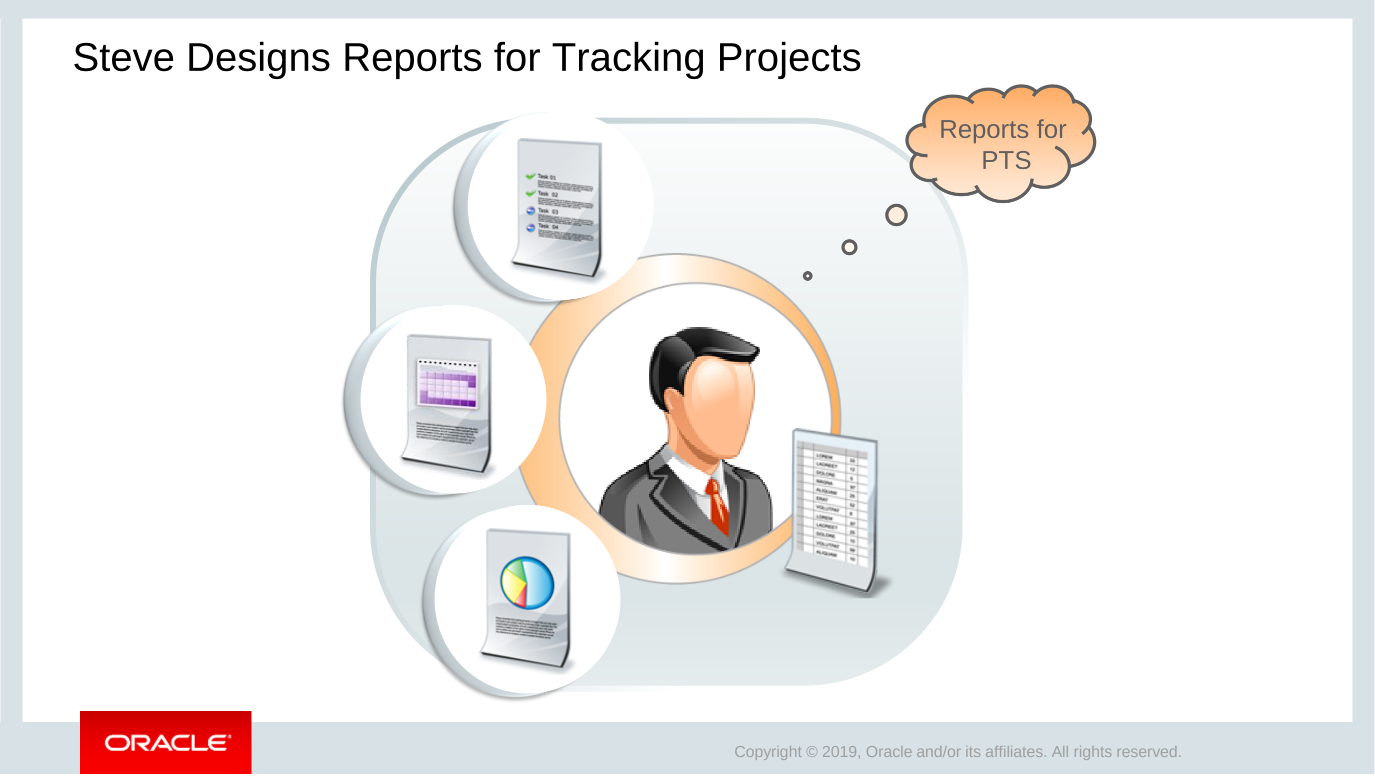


Objectives

After completing this lesson, you should be able to:

- Identify the types of reports that you can create for an application in Oracle Application Express
- Create a Classic report and modify it
- Create and use a List View report
- Create and use a Column Toggle report

This lesson introduces you to the different types of reports for your applications that you create in Oracle Application Express. You are introduced to the various built-in wizards that help you create reports. In this lesson, you will learn how to create and use a Classic report, List View report, and Column Toggle report in particular.



Steve Designs Reports for Tracking Projects

Because Steve already has a good knowledge about how Oracle Application Express works, he now decides to create an application that tracks various aspects of a project. He decides to call it *Project Tracking System* or PTS, in short. As he delves into the requirements, he realizes that PTS should provide users with various reports to bring more clarity in tracking multiple projects. He draws out a couple of reports that PTS should provide.

One report should provide the option to customize all the project details by filtering, sorting, grouping, and so on. He calls it the *Project Master report.*

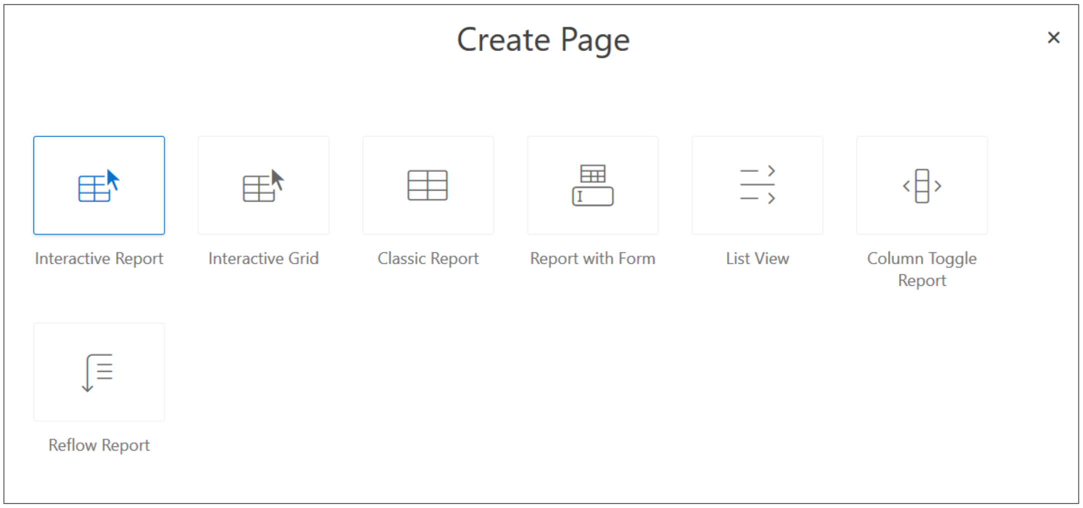
There is also the requirement for tracking project status. This report should provide a list of all the projects with different project statuses and upcoming milestones. He doesn't see the need for any customization at the user's end in this report. He decides to call it the *Project Status Report.*

Another need that Steve envisages is the need to track project documents, such as project plan, data modelling diagrams, schema tracking, technical and functional requirement documents, risk mitigation plan, SQL scripts, and so on, which are used in all projects. By using this report, the end user, who is a project manager in this scenario, should be able to customize the results by searching, filtering, and sorting data based on selected projects. He plans to name it *Project Master Document.*

Lesson Agenda

- Overview
  - Types of Reports
  - Different Ways of Creating a Report
- Creating Classic Reports
- Creating List View Reports
- Creating Column Toggle Reports

Types of Reports



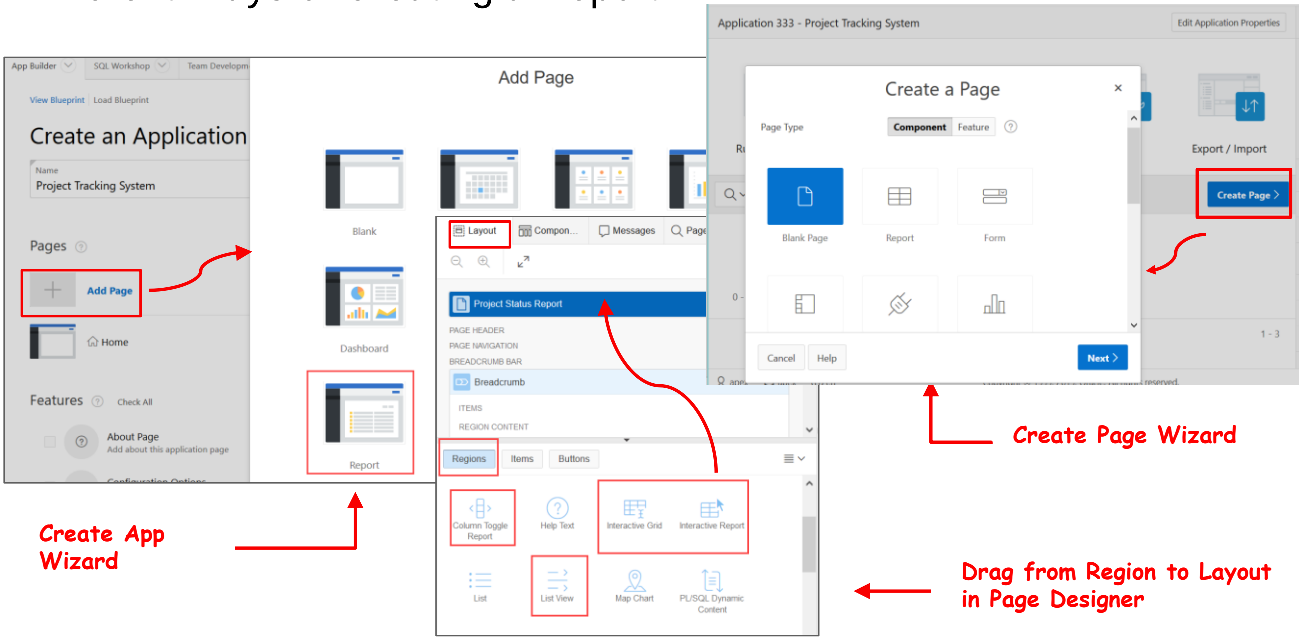
Oracle Application Express offers you a variety of reports with different reporting features and customization options. You can create four basic types of reports: Interactive Grid, Interactive Report, Report with Form on Table, and Classic report.

In this lesson, you learn about Classic reports and two other reports: List View reports and Column Toggle reports. You also learn about the option **Report with Form.**

- **Classic Report:** Creates a report based on a custom SQL `SELECT` statement or a PL/SQL function that returns a SQL `SELECT` statement. It doesn't offer any customization option.
- **Interactive Report:** Creates an interactive report based on a custom SQL `SELECT` statement that you provide. You can customize the layout of their data by selecting the options from the **Actions** menu.
- **Interactive Grid:** Creates an interactive grid and presents the data in a formatted result of a SQL query. You must choose a table on which to build the report or provide a custom SQL query.
- **Report with Form:** Creates an interactive grid report and form based on the table that you specify. For the report, you choose the following report types: Interactive Report, Interactive Grid, or Classic Report.
- **List View Report:** Creates a page that displays the formatted result of a SQL query. This report has a responsive design and is optimal for smartphones and other mobile devices.
- **Column Toggle Report:** Includes a **Columns** button that allows you to select the columns that you want to view. This report also has a responsive design and, therefore, is optimal for smartphones and other mobile devices.

You learn more about Interactive Report and Interactive Grid in lessons titled "Working with Interactive Reports" and "Working with Interactive Grids," respectively. To learn more about **Reflow Report,** refer to the *Oracle Application Express App Builder User's Guide.*

Different Ways of Creating a Report



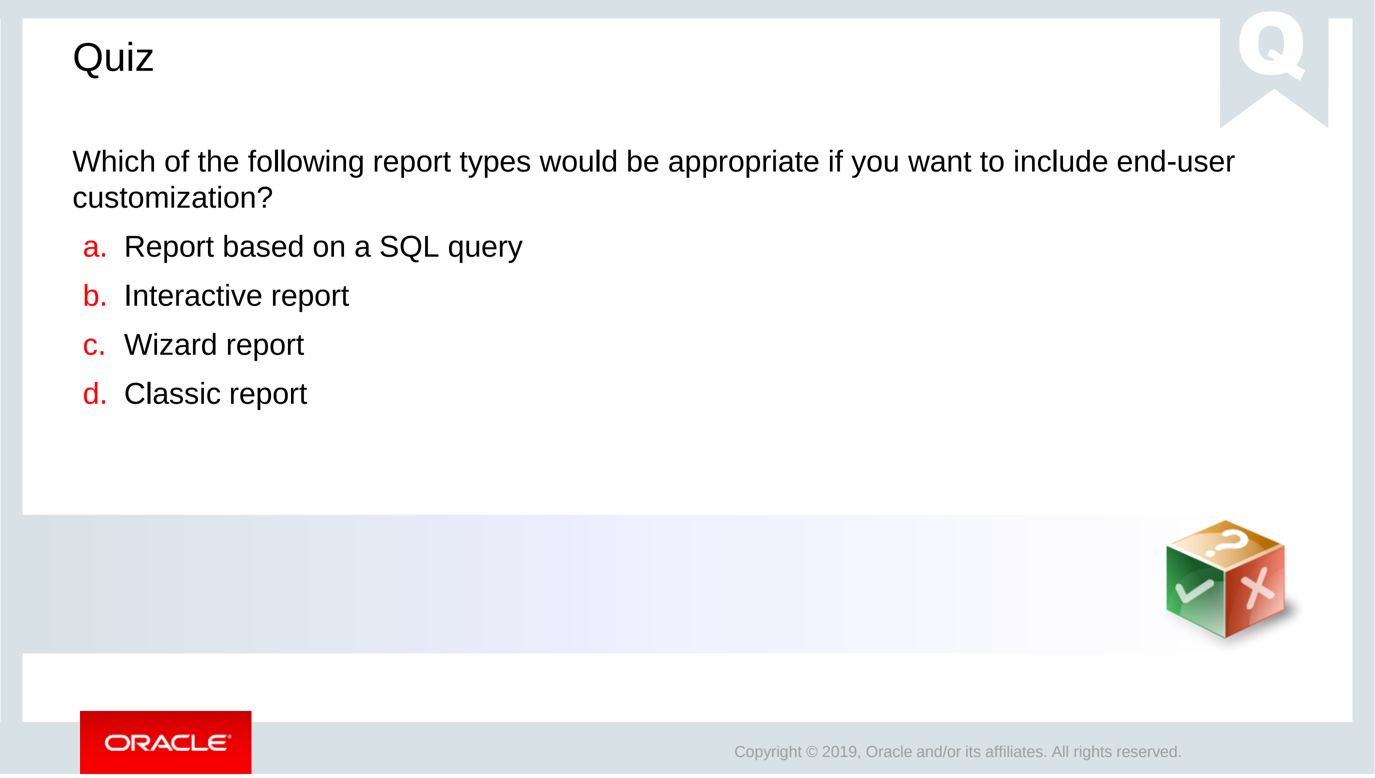
Create App Wizard

Create Page Wizard

Drag from Region to Layout in Page Designer

You can create a report in three different ways:

- By using the **Create App Wizard** while creating an application
  - Supported reports are Interactive Reports, Interactive Grids, and Classic Reports.
- By using the **Create Page** while creating a new page in your application
  - Supported reports are Interactive Report, Interactive Grid, Classic Report, Report with Form on table, List View, Column Toggle Report, Reflow Report, or Report on a Web Service.
- By creating a **Report** region on an existing page using Page Designer. Drag and drop the report type that you want to create from the gallery in the lower central pane to the desired location on the Layout tab.



Quiz

Which of the following report types would be appropriate if you want to include end-user customization?

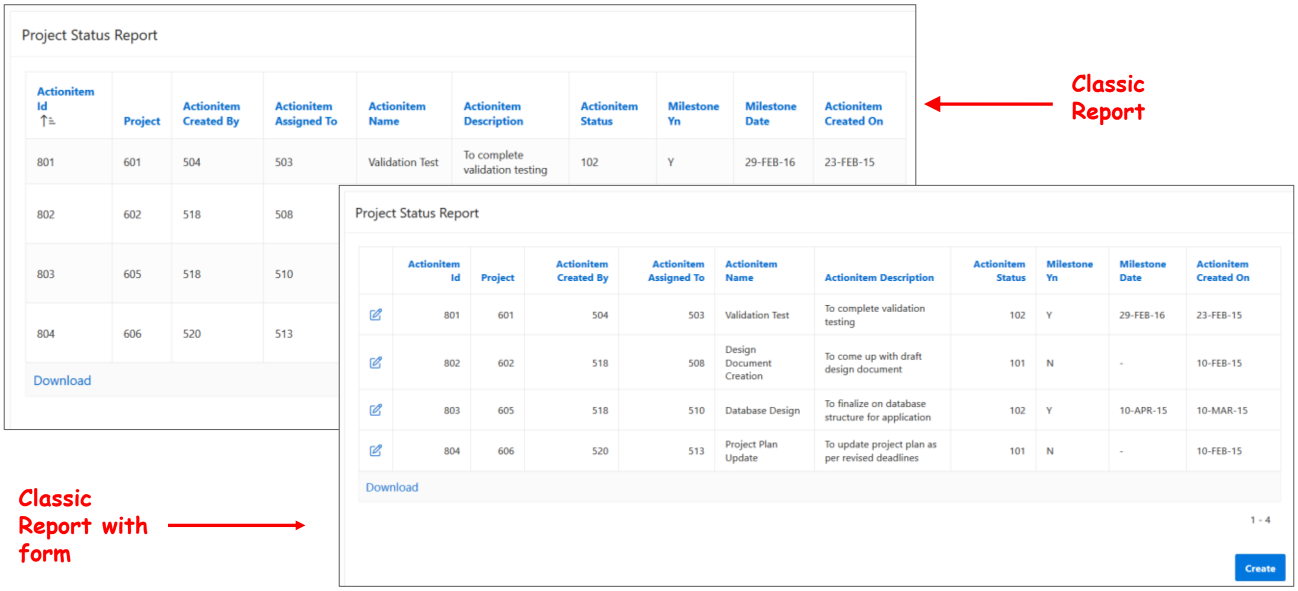
a. Report based on a SQL query
b. Interactive report
c. Wizard report
d. Classic report

Answer: b

Lesson Agenda

- Overview
- Creating Classic Reports
  - Selecting the Appropriate Report Type
  - When Is Classic Report the Best Option?
  - Creating a Classic Report
  - Creating a Classic Report with Form
- Creating List View Reports
- Creating Column Toggle Reports

Selecting the Appropriate Report Type



Project Status Report

Actionitem Id ↑	Project	Actionitem Created By	Actionitem Assigned To	Actionitem Name	Actionitem Description	Actionitem Status	Milestone Yn	Milestone Date	Actionitem Created On
801	601	504	503	Validation Test	To complete validation testing	102	Y	29-FEB-16	23-FEB-15
802	602	518	508						
803	605	518	510						
804	606	520	513						

Download

Classic Report

Project Status Report

	Actionitem Id	Project	Actionitem Created By	Actionitem Assigned To	Actionitem Name	Actionitem Description	Actionitem Status	Milestone Yn	Milestone Date	Actionitem Created On
✎	801	601	504	503	Validation Test	To complete validation testing	102	Y	29-FEB-16	23-FEB-15
✎	802	602	518	508	Design Document Creation	To come up with draft design document	101	N	-	10-FEB-15
✎	803	605	518	510	Database Design	To finalize on database structure for application	102	Y	10-APR-15	10-MAR-15
✎	804	606	520	513	Project Plan Update	To update project plan as per revised deadlines	101	N	-	10-FEB-15

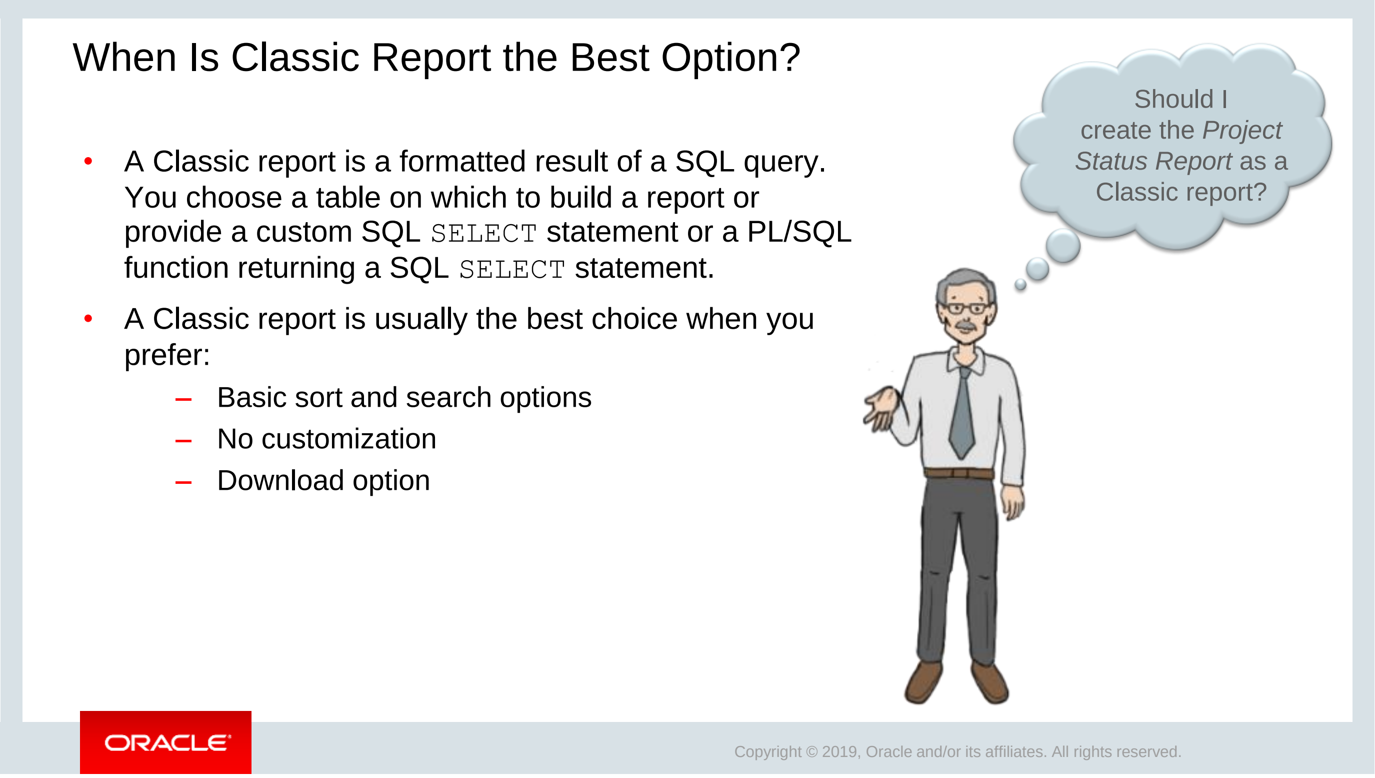
Download

1 - 4

Classic Report with form

Before we proceed with Classic reports, let us quickly look at the major difference between a Classic report and a Classic report with form. The examples in the slide show a Classic report and a Classic Report with form. Both these reports are created using the Create Page Wizard. The Classic report queries the same columns in the `PROJECTS_ACTIONITEMS` table as the Classic report with form. The following is the only difference between the two:

- **Classic report with form:** This report provides a snapshot of all the details pulled from the `PROJECTS_ACTIONITEMS` table along with a link to the form that contains the details of each row. By clicking the pencil icon, you can edit the values for each row in the report. You can download this report and also create a new entry by using the **Create** option. However, you cannot customize this report.
- **Classic report:** This report also provides a snapshot of all the details pulled from the `PROJECTS_ACTIONITEMS` table, but with only basic sorting, keyword search, and download options. You cannot edit or customize this report.



When Is Classic Report the Best Option?

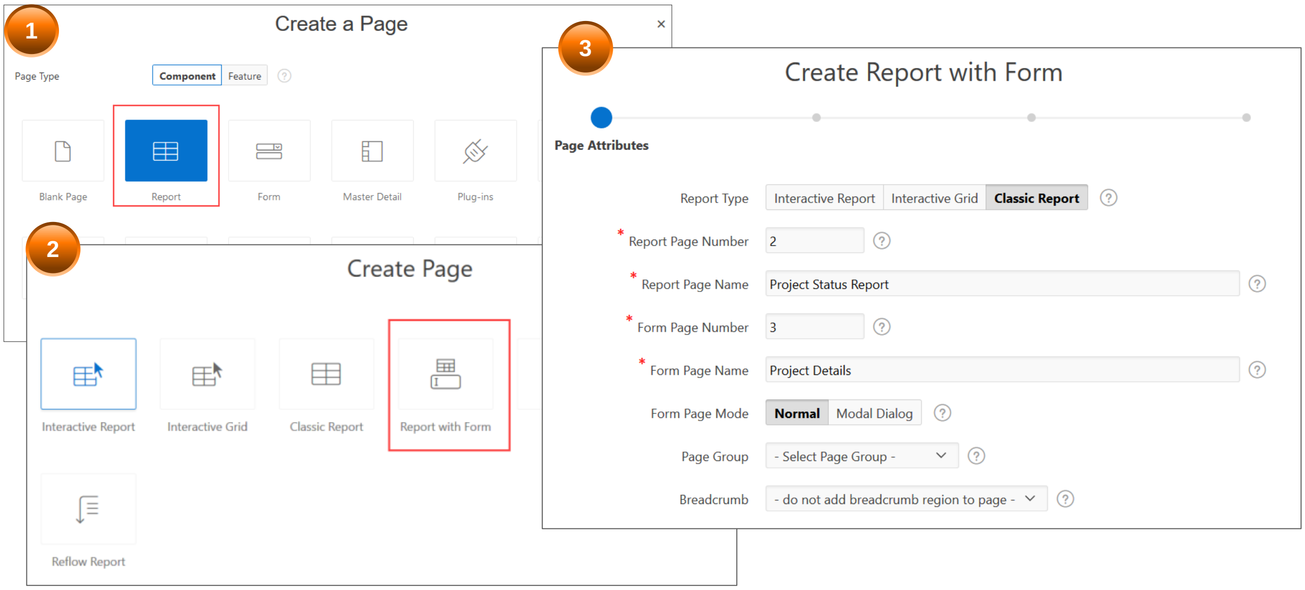
- A Classic report is a formatted result of a SQL query. You choose a table on which to build a report or provide a custom SQL `SELECT` statement or a PL/SQL function returning a SQL `SELECT` statement.
- A Classic report is usually the best choice when you prefer:
  - Basic sort and search options
  - No customization
  - Download option

Should I create the *Project Status Report* as a Classic report?

One of the reports that Steve wants to provide in the PTS application is a report that provides a snapshot of all the action items along with their current status for all the projects managed by a project manager. The report should have basic search and sort options, without customization.

Steve feels that a Classic report would be the best fit for this requirement. He creates the *Project Status Report* as a Classic report in the PTS application.

Creating Classic Report with Form



Report Type: Interactive Report | Interactive Grid | **Classic Report**

* Report Page Number: 2

* Report Page Name: Project Status Report

* Form Page Number: 3

* Form Page Name: Project Details

Form Page Mode: **Normal** | Modal Dialog

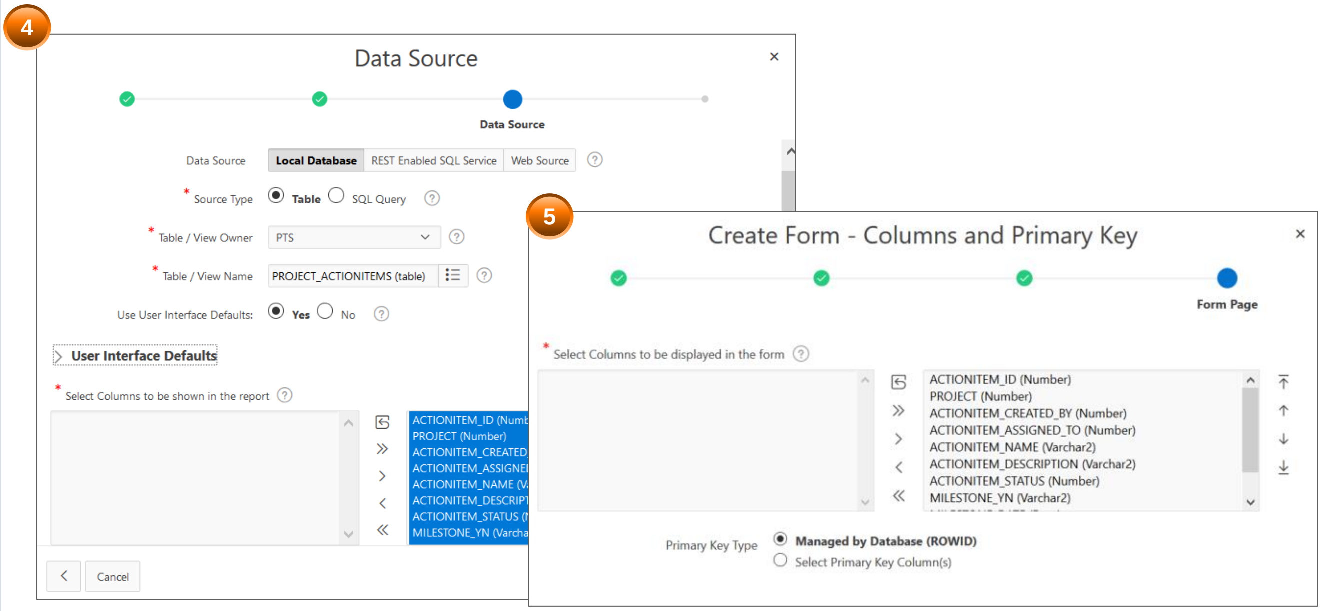
Page Group: - Select Page Group -

Breadcrumb: - do not add breadcrumb region to page -

The slide shows the steps to create a Classic report with Form. To create a Classic report with Form:

1. Access the Create Report Wizard by clicking the **Create Page** button on the Application home page.
2. Select **Report** for the page type.
3. Select **Report with Form** and click **Next** (screenshot 1).
4. In the Create Report with Form dialog, define the following and click **Next** (screenshot 2):
   - **a) Report Type:** Select **Classic Report**
   - **b) Report Page Number:** `2`
   - **c) Report Page Name:** Enter `Project Status Report`
   - **d) Form Page number:** `3`
   - **e) Form Name:** Enter `Project Members`
   - **f) Form Page mode:** Select **Normal**
5. Specify the following settings for navigation menu, if you want to add it to the navigation menu, and click **Next:**
   - **a) Navigation Preference:** Select `Create a new navigation menu entry`
   - **b) New Navigation Menu Entry:** `Project Status Report`
   - **c) Parent Navigation Menu Entry:** Select `– no parent selected-`

Creating Classic Reports with Form



4

Data Source

Data Source

Data Source: Local Database | REST Enabled SQL Service | Web Source

* Source Type: ● Table ○ SQL Query

* Table / View Owner: PTS

* Table / View Name: PROJECT_ACTIONITEMS (table)

Use User Interface Defaults: ● Yes ○ No

User Interface Defaults

* Select Columns to be shown in the report

- ACTIONITEM_ID (Numb
- PROJECT (Number)
- ACTIONITEM_CREATED
- ACTIONITEM_ASSIGNE
- ACTIONITEM_NAME (V
- ACTIONITEM_DESCRIPT
- ACTIONITEM_STATUS (
- MILESTONE_YN (Varcha

5

Create Form - Columns and Primary Key

Form Page

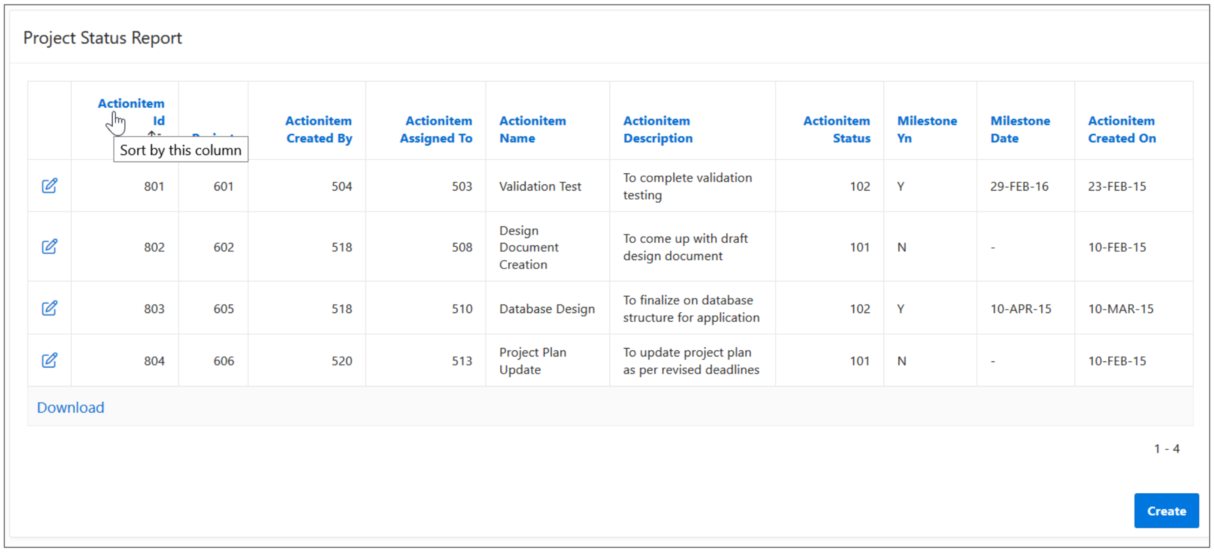
* Select Columns to be displayed in the form

- ACTIONITEM_ID (Number)
- PROJECT (Number)
- ACTIONITEM_CREATED_BY (Number)
- ACTIONITEM_ASSIGNED_TO (Number)
- ACTIONITEM_NAME (Varchar2)
- ACTIONITEM_DESCRIPTION (Varchar2)
- ACTIONITEM_STATUS (Number)
- MILESTONE_YN (Varchar2)

Primary Key Type: ● Managed by Database (ROWID) ○ Select Primary Key Column(s)

6. For the report data source, click **Local Database** and define the following:
   - **Source Type:** Select **Table**
   - **Table/View Owner:** `PTS` (selected by default, since you are working with the PTS schema)
   - **Table/View Name:** Select `PROJECT_ACTIONITEMS (table)`
   - **Use User Interface Default:** Yes (default)
   - Click >> to include all columns in the report
   - Click **Next.**
7. For Form Page, define the following:
   - Primary Key Type: `Managed by Database (ROWID)`
   - Click **>>** to include all columns in the form
8. Click **Create.** The page is now created, and it opens in Page Designer view.
9. In Page Designer, click **Report 1** under Content Body in the Rendering tab. In Property Editor, under Identification, edit the **Title** to `Project Status Report.`
10. Click **Save and Run.**

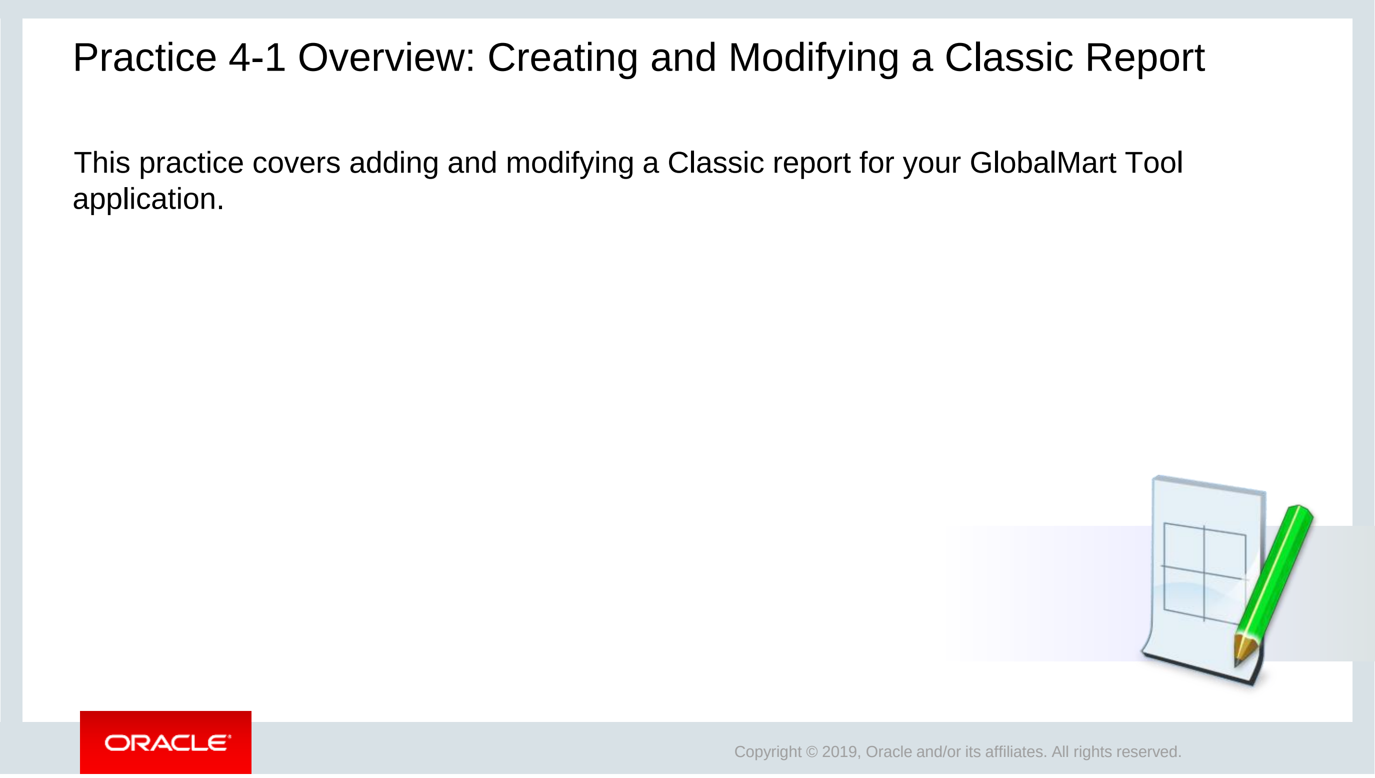
Classic Report



Project Status Report

	Actionitem Id		Actionitem Created By	Actionitem Assigned To	Actionitem Name	Actionitem Description	Actionitem Status	Milestone Yn	Milestone Date	Actionitem Created On
✎	801	601	504	503	Validation Test	To complete validation testing	102	Y	29-FEB-16	23-FEB-15
✎	802	602	518	508	Design Document Creation	To come up with draft design document	101	N	-	10-FEB-15
✎	803	605	518	510	Database Design	To finalize on database structure for application	102	Y	10-APR-15	10-MAR-15
✎	804	606	520	513	Project Plan Update	To update project plan as per revised deadlines	101	N	-	10-FEB-15

The *Project Status Report*, which is a Classic report with form, is generated. It has the option to download the report. You can click the column names to sort the columns. You can click the edit icon to view and edit the details in the Project Details form.



Practice 4-1 Overview: Creating and Modifying a Classic Report

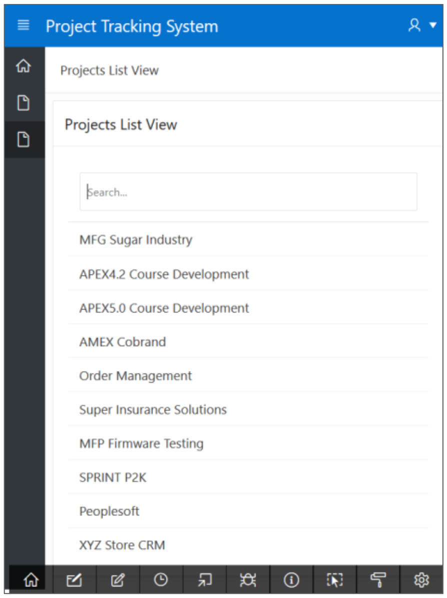
This practice covers adding and modifying a Classic report for your GlobalMart Tool application.

List View Reports

There are managers who prefer to access project reports on their mobile devices. To cater to such requirements, you can consider creating List View reports.

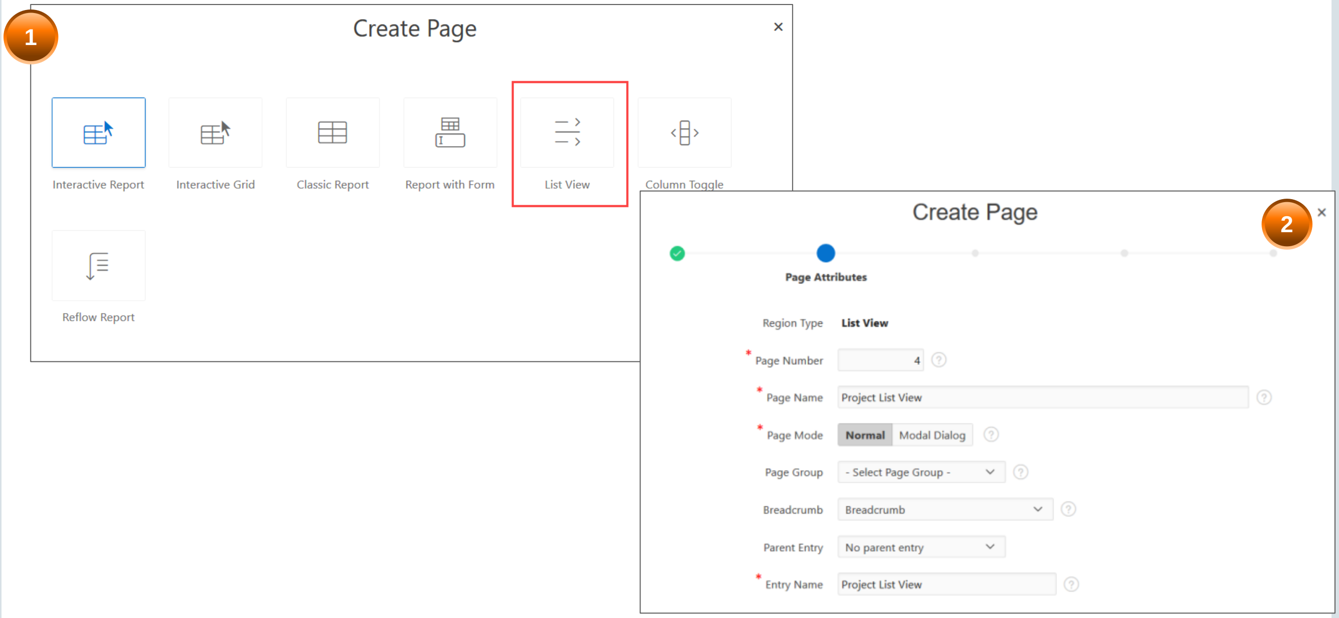
List View reports carry a responsive design and is best suited when you have to display data and provide easy navigation on smartphones and other mobile devices.

- It creates a page that contains the formatted result of a SQL query.
- You choose a table on which to build the List view and select a database column to be used for the List view entry.



MFG Sugar Industry
APEX4.2 Course Development
APEX5.0 Course Development
AMEX Cobrand
Order Management
Super Insurance Solutions
MFP Firmware Testing
SPRINT P2K
Peoplesoft
XYZ Store CRM

Creating a List View Report



Region Type List View

* Page Number 4

* Page Name Project List View

* Page Mode Normal Modal Dialog

Page Group - Select Page Group -

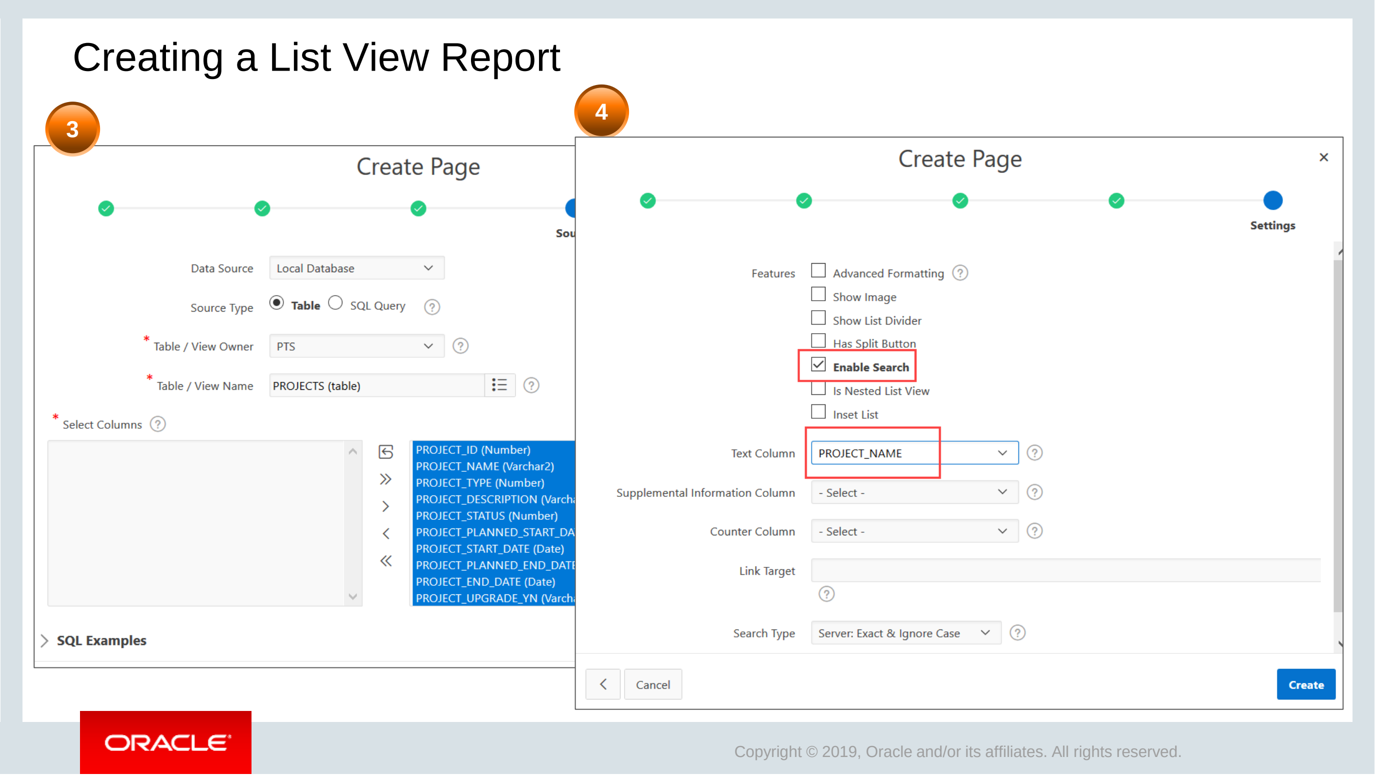
Breadcrumb Breadcrumb

Parent Entry No parent entry

* Entry Name Project List View

To create a List View, perform the following steps:

1. Access the **Create Page** Wizard and select **Report.**
2. Select **List View** for the report type (screenshot 1).
3. For Page Attributes, specify the Page Name and select Breadcrumbs. The Page Number and Entry name (same as Page Name) are allotted by default (screenshot 2). Click **Next.**
4. For Navigation menu, select **Create a New Navigation Menu Entry**. Select Home for the **Parent Navigation Menu Entry.**



Creating a List View Report

Create Page

Data Source: Local Database

Source Type: ● Table ○ SQL Query

* Table / View Owner: PTS

* Table / View Name: PROJECTS (table)

* Select Columns

PROJECT_ID (Number)
PROJECT_NAME (Varchar2)
PROJECT_TYPE (Number)
PROJECT_DESCRIPTION (Varch
PROJECT_STATUS (Number)
PROJECT_PLANNED_START_DA
PROJECT_START_DATE (Date)
PROJECT_PLANNED_END_DATE
PROJECT_END_DATE (Date)
PROJECT_UPGRADE_YN (Varch

Create Page

Features:
□ Advanced Formatting
□ Show Image
□ Show List Divider
□ Has Split Button
☑ Enable Search
□ Is Nested List View
□ Inset List

Text Column: PROJECT_NAME

Supplemental Information Column: - Select -

Counter Column: - Select -

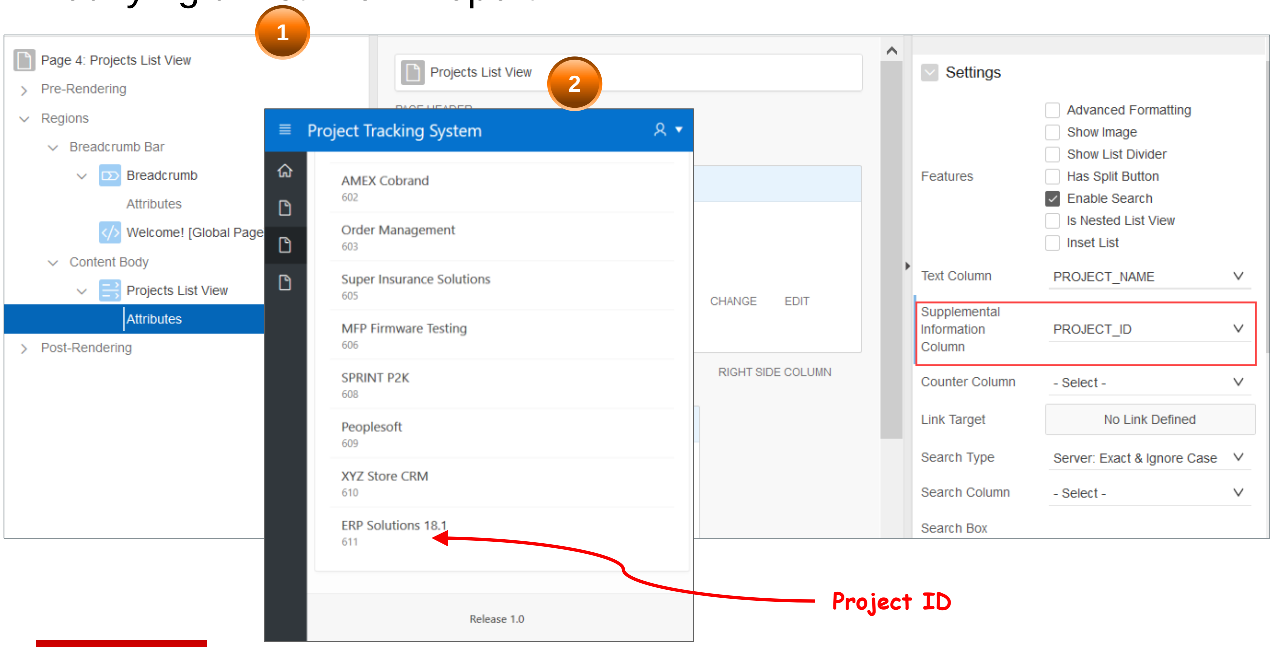
Link Target:

Search Type: Server: Exact & Ignore Case

This slide shows the remaining steps for creating a List View report and a sample List View report snapshot.

5. For the report data source, specify the table (screenshot 4).
6. Specify the Report Settings (such as features to enable search and search column in the report).
7. Click **Create** (screenshot 5).

Modifying a List View Report

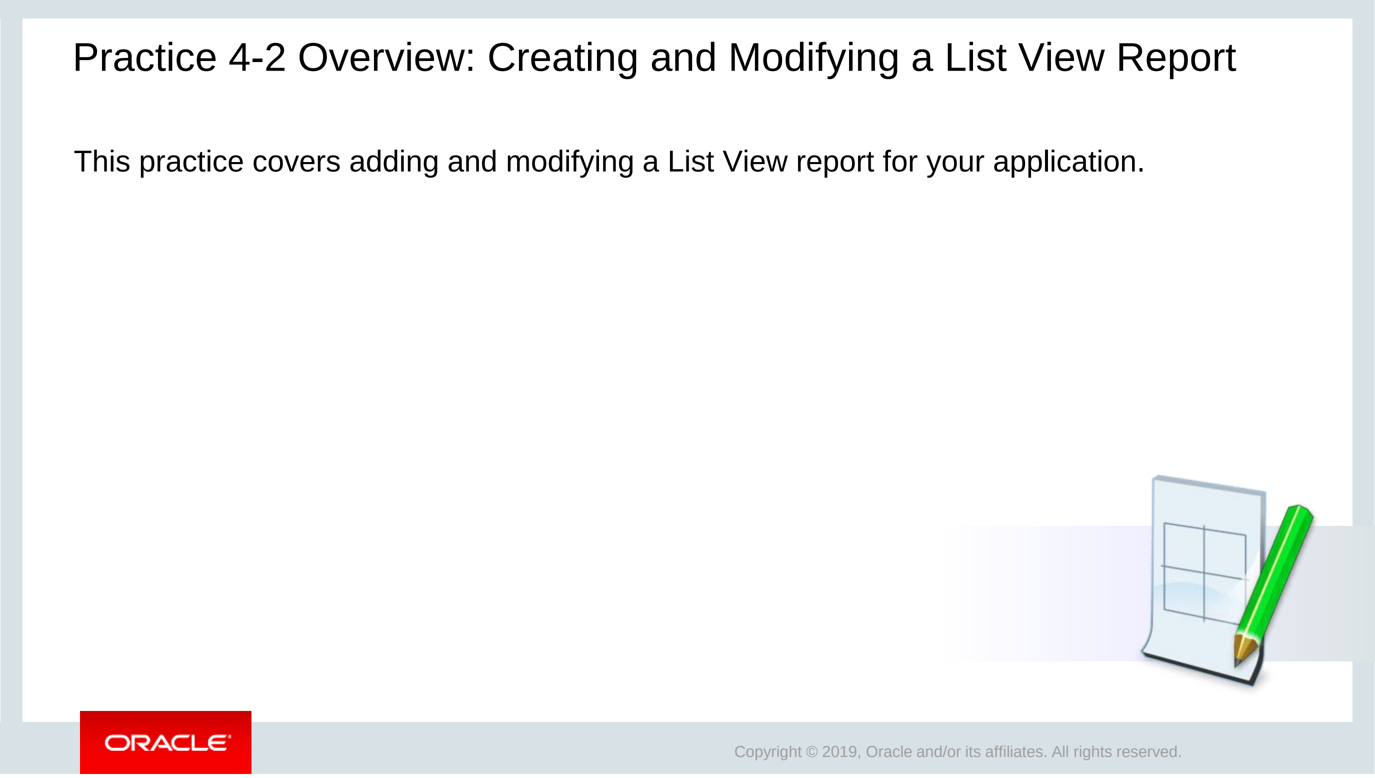


AMEX Cobrand	602
Order Management	603
Super Insurance Solutions	605
MFP Firmware Testing	606
SPRINT P2K	608
Peoplesoft	609
XYZ Store CRM	610
ERP Solutions 18.1	611

Settings

Features	□ Advanced Formatting □ Show Image □ Show List Divider □ Has Split Button ☑ Enable Search □ Is Nested List View □ Inset List
Text Column	PROJECT_NAME
Supplemental Information Column	PROJECT_ID
Counter Column	- Select -
Link Target	No Link Defined
Search Type	Server: Exact & Ignore Case
Search Column	- Select -
Search Box	

You can modify your List View report by modifying the features and some attributes of the report. For example, selecting Enable Search allows you to enable search in your report based on the Search Column. Similarly, adding Supplemental Information Column enables you to add supplemental information for the List View entry. In the example shown in the slide, project ID is the Supplemental Information Column using Page Designer Property Editor.



Practice 4-2 Overview: Creating and Modifying a List View Report

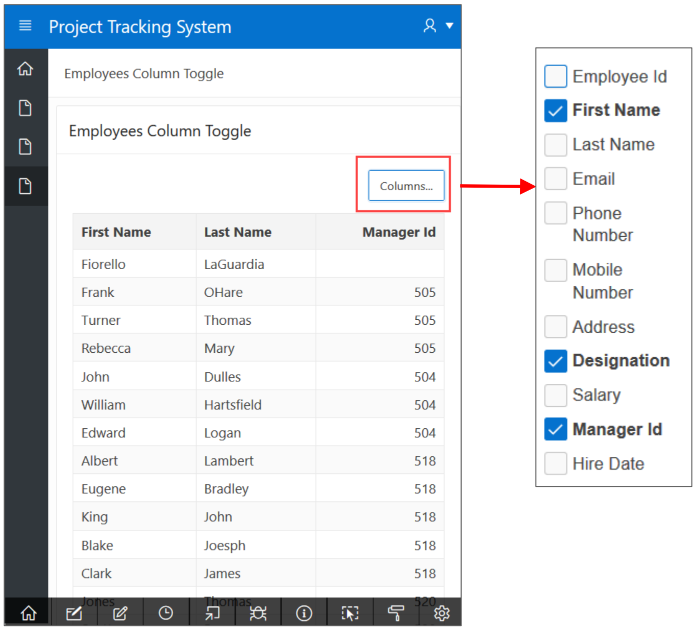
This practice covers adding and modifying a List View report for your application.

Column Toggle Reports

For some reports, users may prefer to view only the relevant columns and not all that the report provides. In this scenario, the Column Toggle report is the best fit.

The Column Toggle report enables you to see the entire or partial record depending on which columns you select or deselect from the toggle list.

- You choose a table on which to build the Column Toggle report.
- The Column Toggle report is optimized for use in a mobile device.



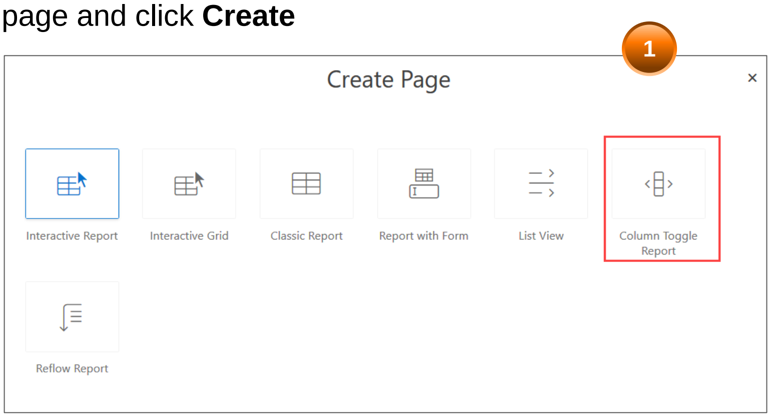
First Name	Last Name	Manager Id
Fiorello	LaGuardia	
Frank	OHare	505
Turner	Thomas	505
Rebecca	Mary	505
John	Dulles	504
William	Hartsfield	504
Edward	Logan	504
Albert	Lambert	518
Eugene	Bradley	518
King	John	518
Blake	Joesph	518
Clark	James	518
Jones	Thomas	520

- □ Employee Id
- ☑ First Name
- □ Last Name
- □ Email
- □ Phone Number
- □ Mobile Number
- □ Address
- ☑ Designation
- □ Salary
- ☑ Manager Id
- □ Hire Date

Creating a Column Toggle Report

To create a Column Toggle Report from Application Builder:

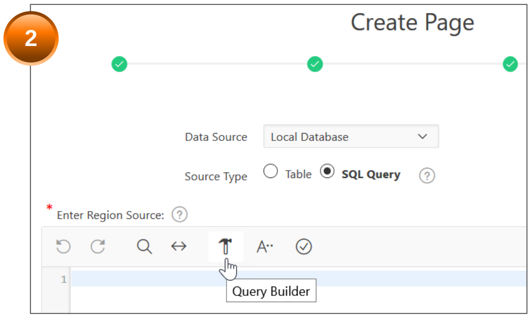
1. Navigate to the Application home page and click **Create Page.**
2. Select **Report.**
3. For Report Type, select **Column Toggle Report.**
4. Specify the Page Attributes and click Next:
   - Page Name – `Employees Column Toggle Report`
   - Page number - `5`
   - Select **Breadcrumbs.**
5. Select **Create a New Navigation Entry** and click N**ext.**



This slide shows the steps to create a Column Toggle report. This type of report can be created on a database application.

Creating a Column Toggle Report

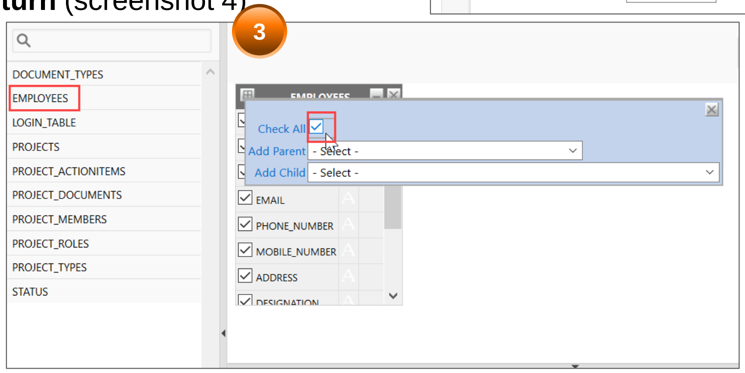
6. For the data source of the report, click **SQL**.
   – Click the **Query Builder** icon (screenshot 2) and select the `EMPLOYEES` table (screenshot 3).
   – Validate the query and click **Return** (screenshot 4).
7. Click **Create.**

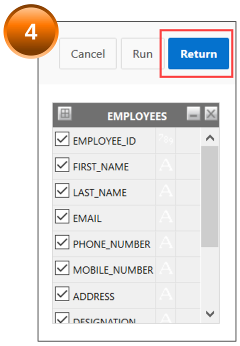


Data Source: Local Database

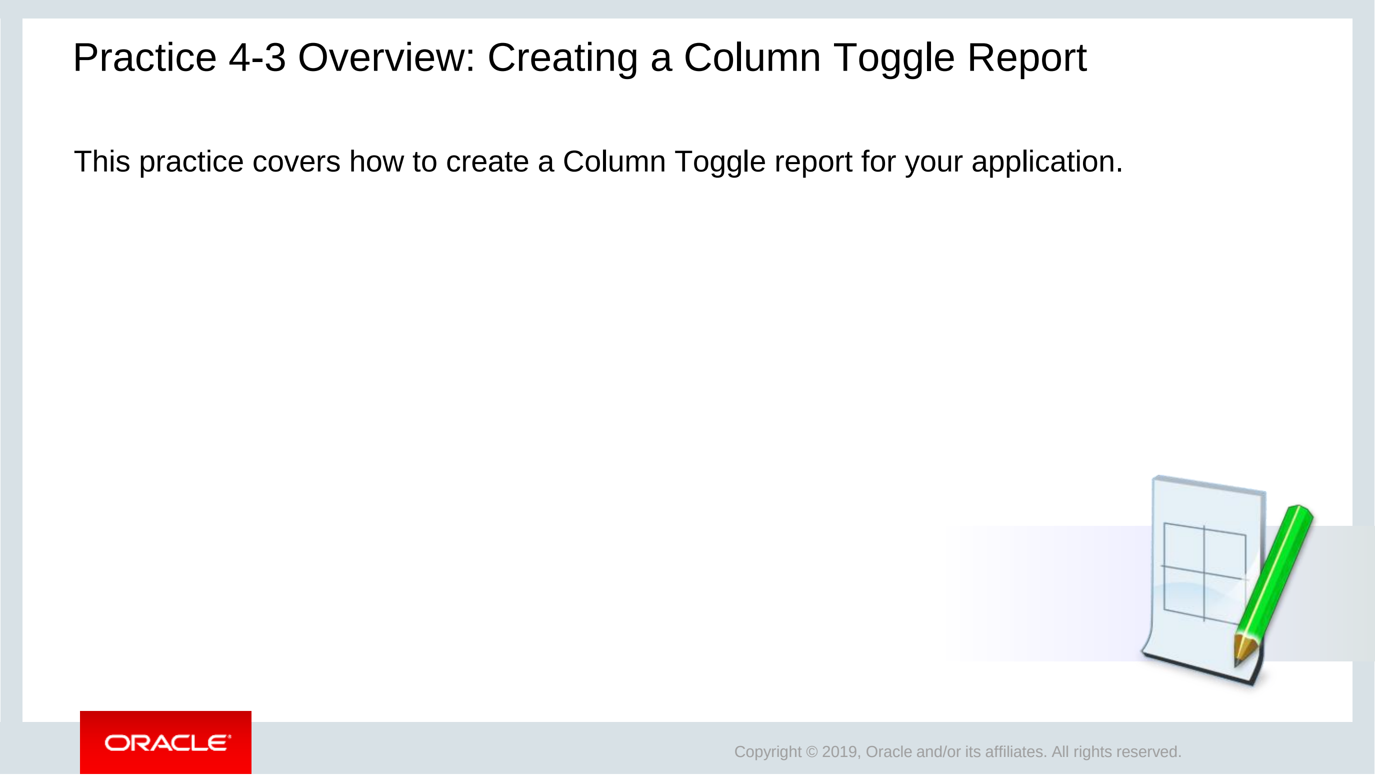
Source Type: ○ Table ● SQL Query

* Enter Region Source:



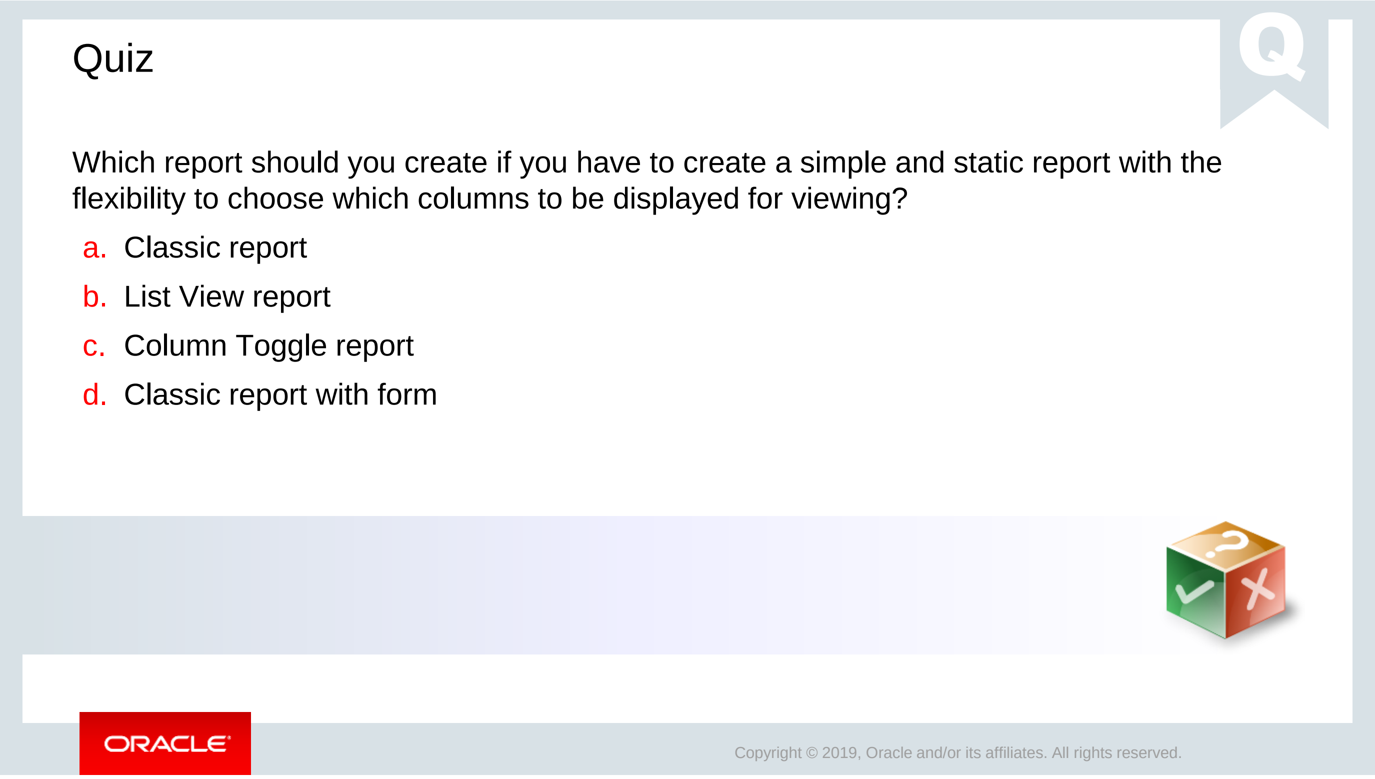


This slide shows the steps to create a Column Toggle report. This type of report can be created on a database application.



Practice 4-3 Overview: Creating a Column Toggle Report

This practice covers how to create a Column Toggle report for your application.



Quiz

Which report should you create if you have to create a simple and static report with the flexibility to choose which columns to be displayed for viewing?

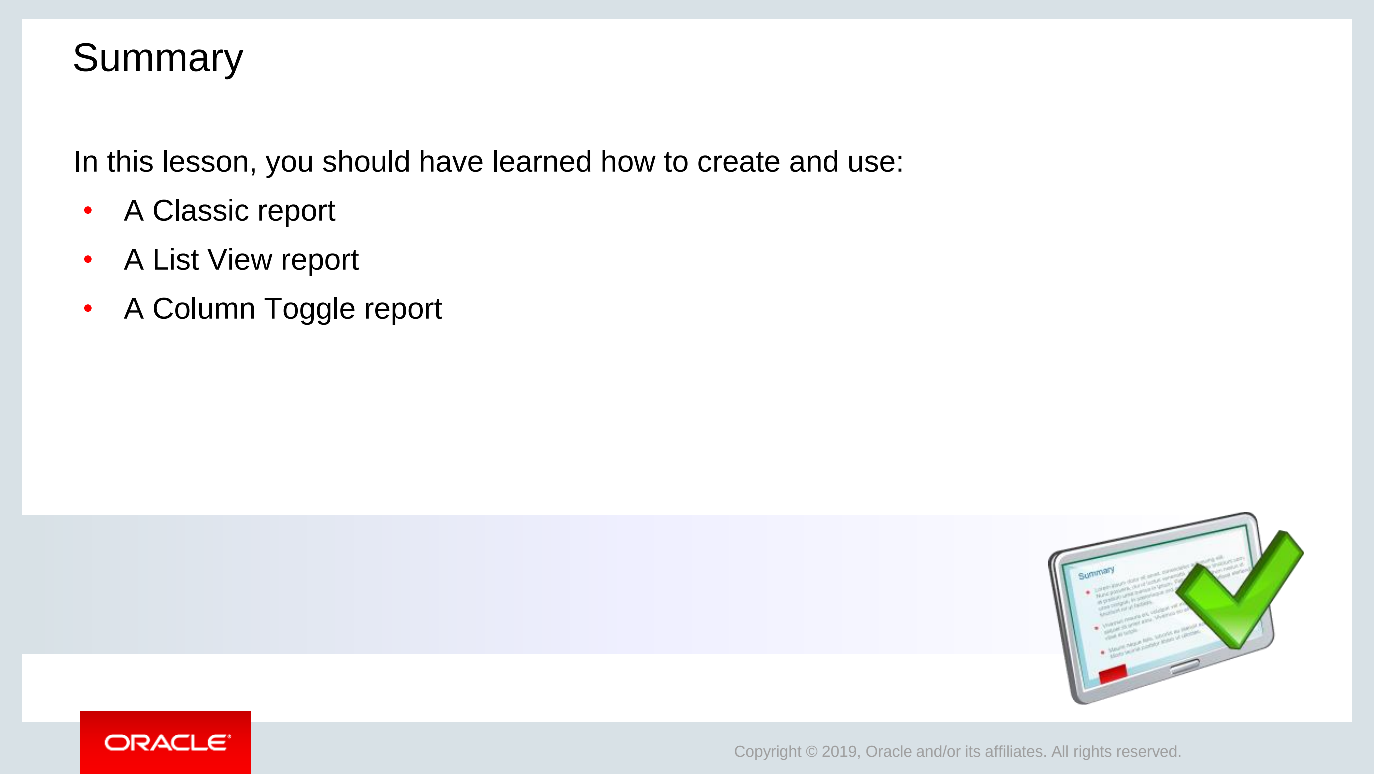
a. Classic report

b. List View report

c. Column Toggle report

d. Classic report with form

Answer: c



Summary

In this lesson, you should have learned how to create and use:

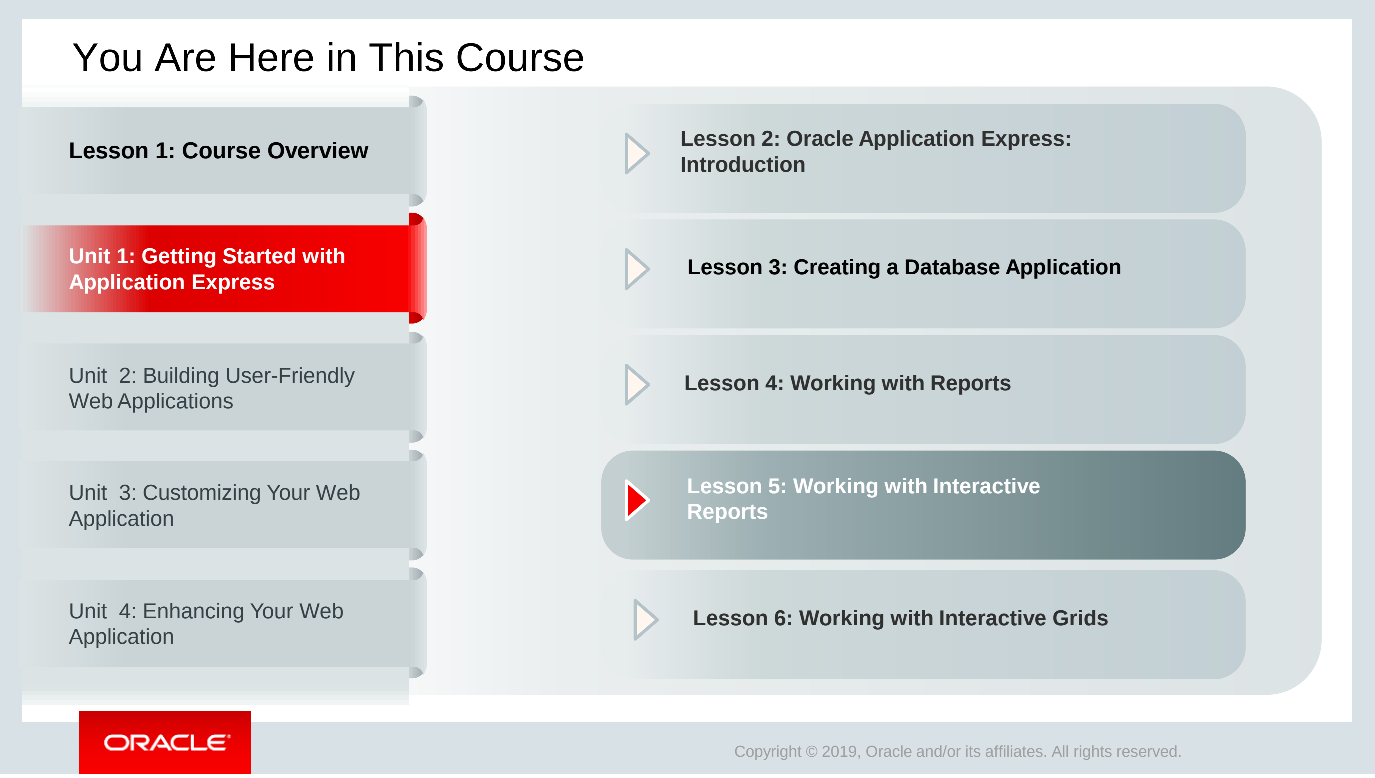
- A Classic report
- A List View report
- A Column Toggle report

ORACLE

In this lesson, you learned how to create Classic (SQL), List View, and Column Toggle reports.

5

Working with Interactive Reports



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

Lesson 2: Oracle Application Express: Introduction

Lesson 3: Creating a Database Application

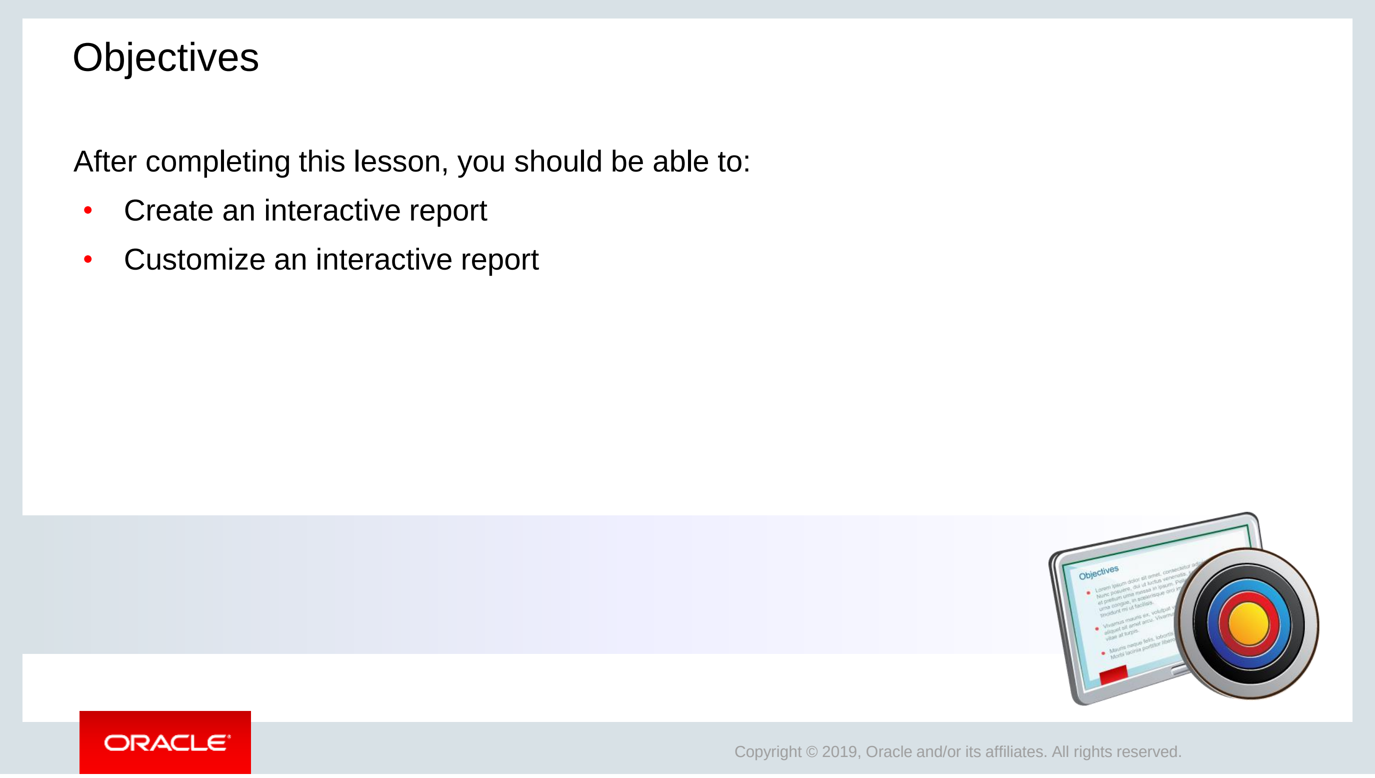
Lesson 4: Working with Reports

Lesson 5: Working with Interactive Reports

Lesson 6: Working with Interactive Grids

ORACLE

This slide shows a graphical depiction of the course, highlighting Unit 1 – Lesson 5 in particular, which is dealt with in these slides.

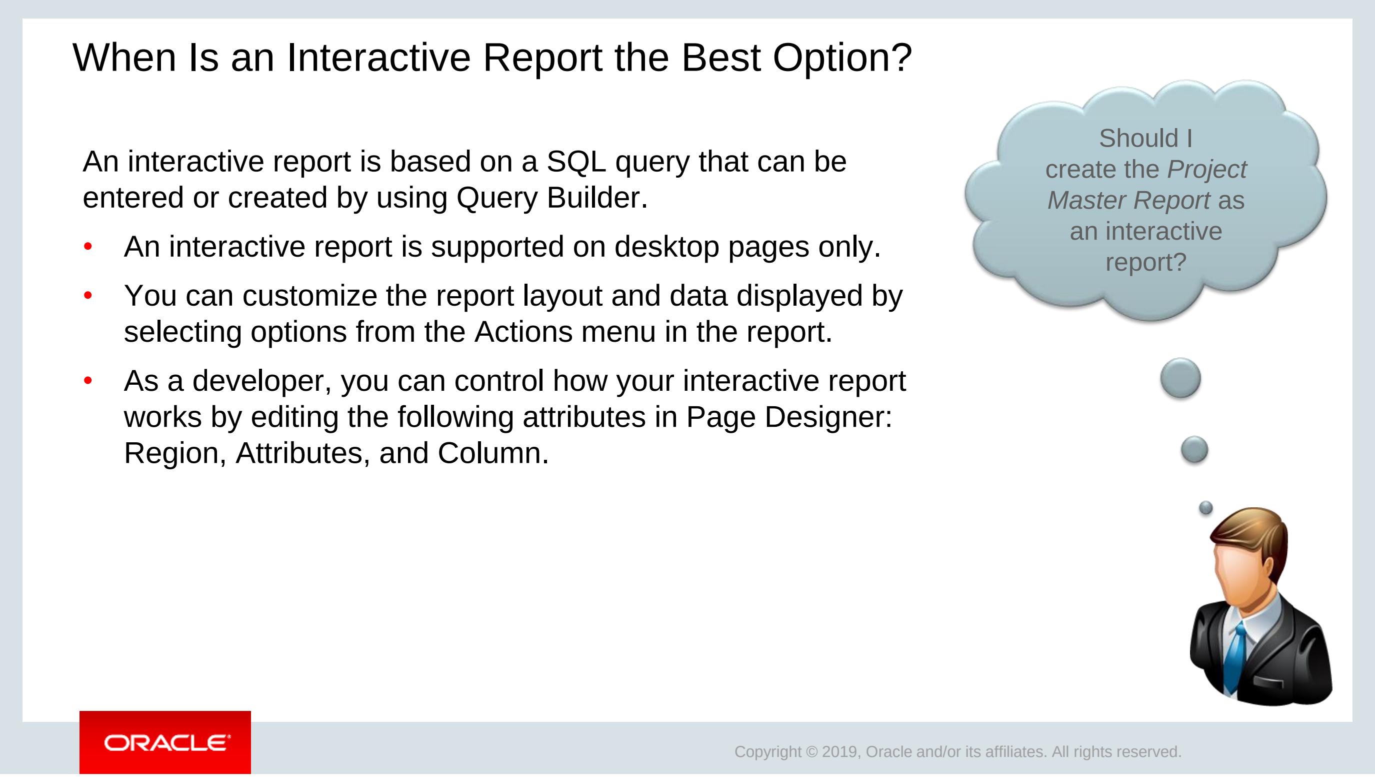


Objectives

After completing this lesson, you should be able to:

- Create an interactive report
- Customize an interactive report

This lesson introduces you to interactive reports in Oracle Application Express. You learn how to create and customize an interactive report.



When Is an Interactive Report the Best Option?

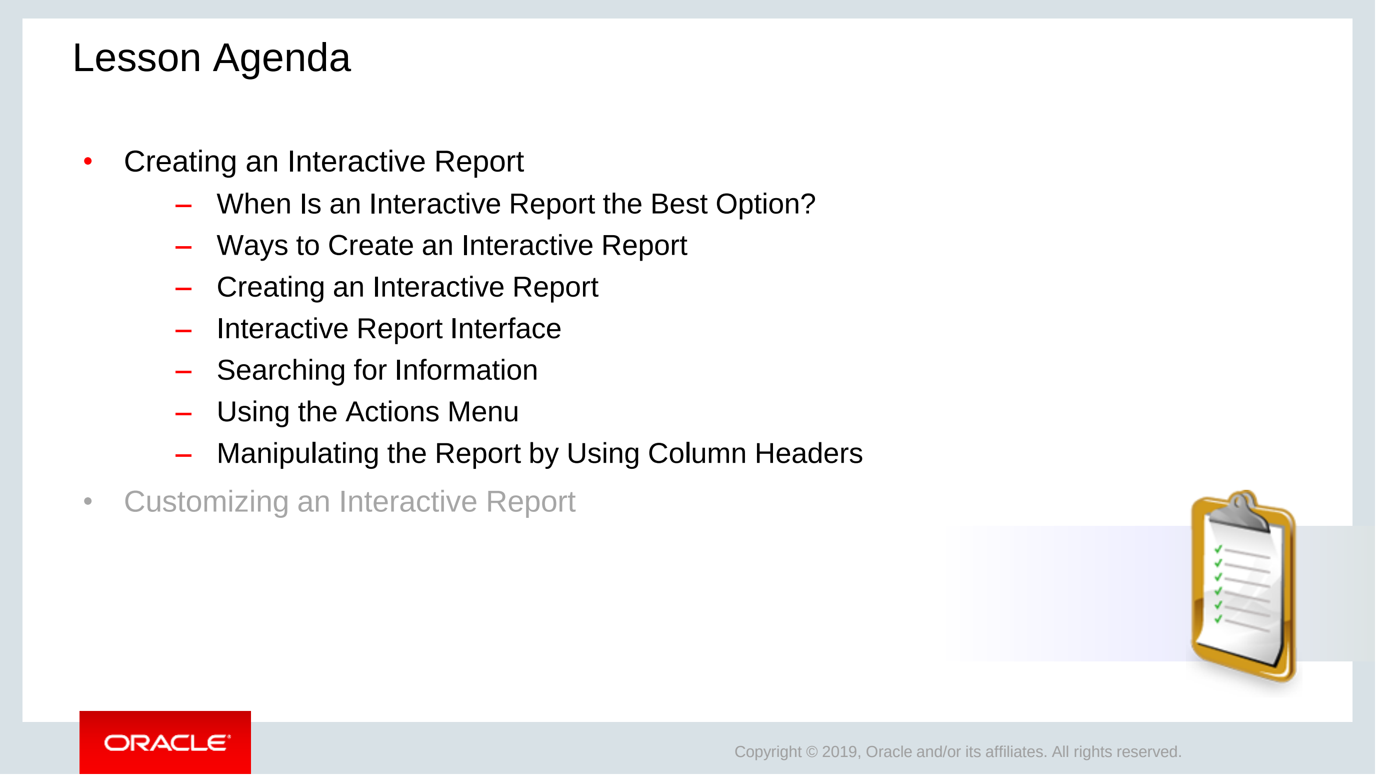
An interactive report is based on a SQL query that can be entered or created by using Query Builder.

- An interactive report is supported on desktop pages only.
- You can customize the report layout and data displayed by selecting options from the Actions menu in the report.
- As a developer, you can control how your interactive report works by editing the following attributes in Page Designer: Region, Attributes, and Column.

Should I create the *Project Master Report* as an interactive report?

Another report that Steve wants in the PTS application is the *Project Master Report.*

For Steve, the requirement is that the report should present complete details of all the projects managed by PTS, such as project ID, project name, project type, status, project start and end dates – both actual and planned dates, project started by, project updated by, and so on. Also, if the report is interactive, then it would allow the project managers to manipulate the report and see only those details in which they are interested in and for only those projects that they manage. Therefore, Steve decides to create an interactive report and calls it the *Project Master Report*.



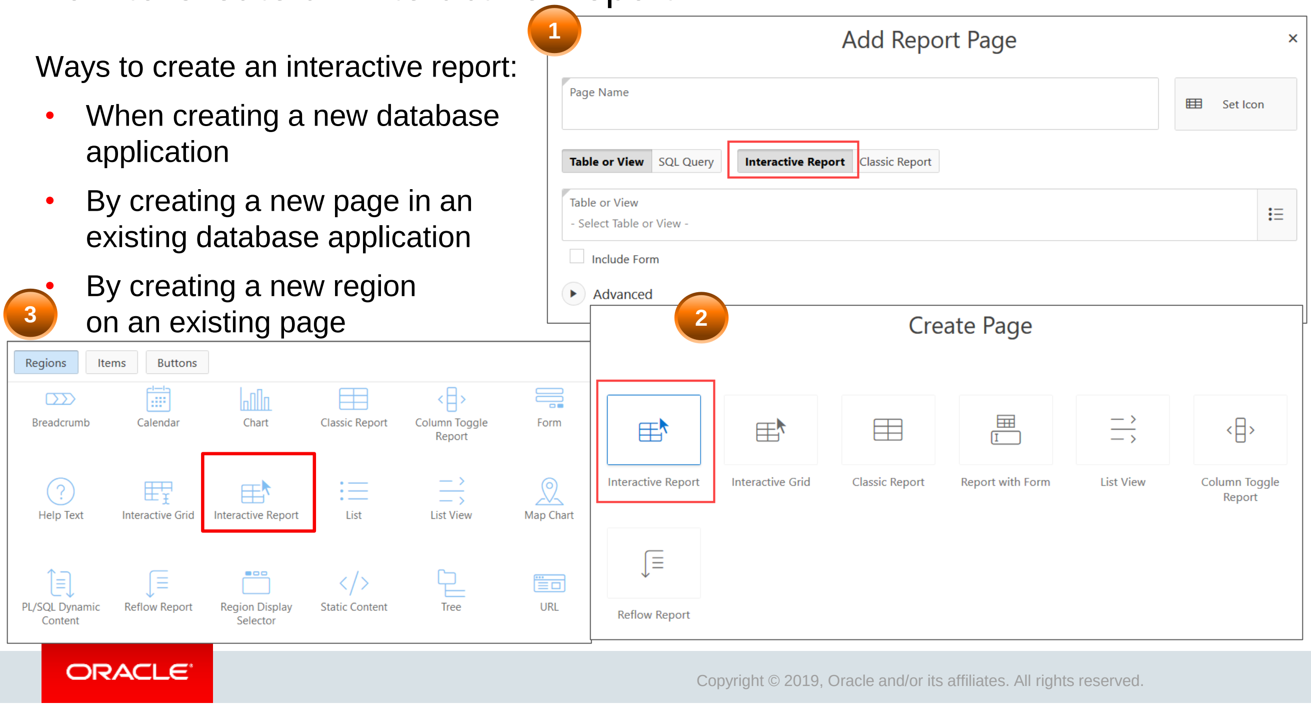
Lesson Agenda

- Creating an Interactive Report
  - When Is an Interactive Report the Best Option?
  - Ways to Create an Interactive Report
  - Creating an Interactive Report
  - Interactive Report Interface
  - Searching for Information
  - Using the Actions Menu
  - Manipulating the Report by Using Column Headers
- Customizing an Interactive Report

How to Create an Interactive Report

Ways to create an interactive report:

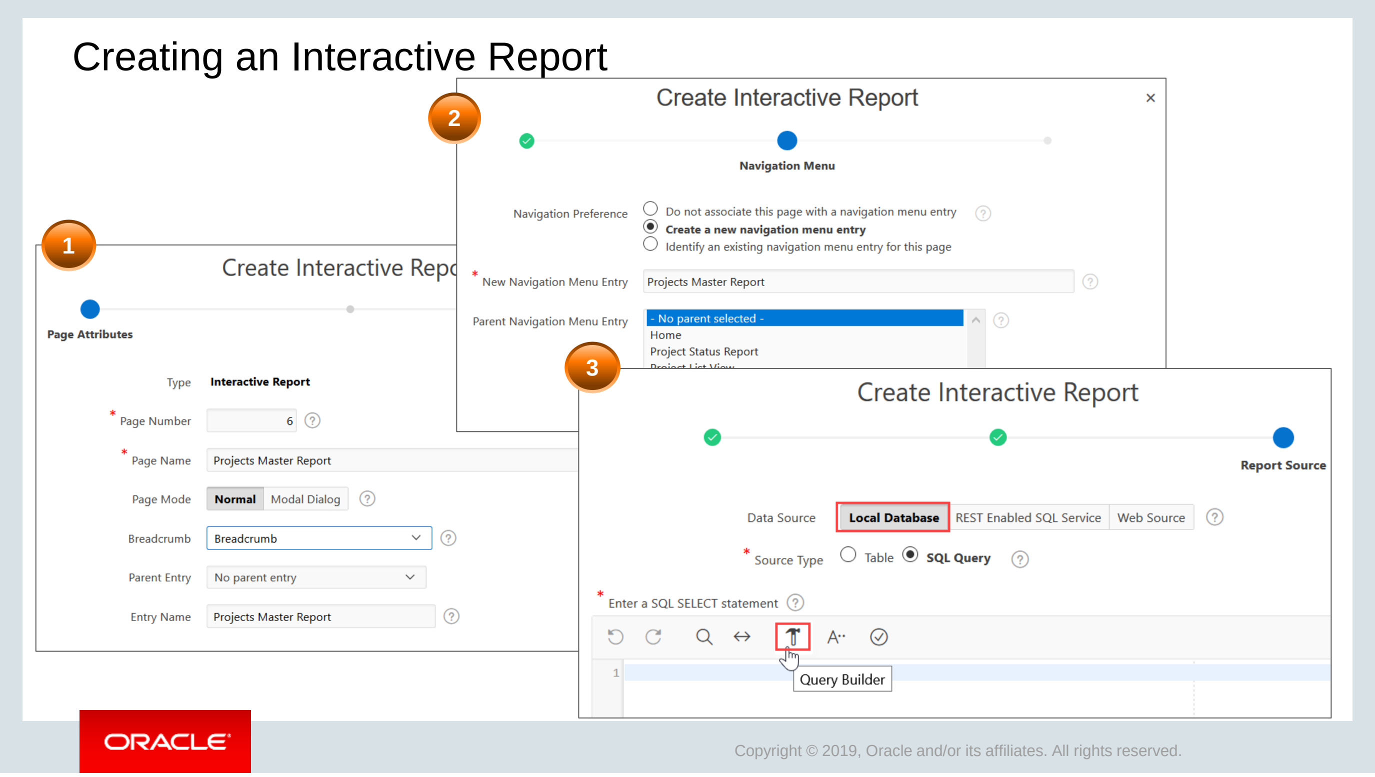
- When creating a new database application
- By creating a new page in an existing database application
- By creating a new region on an existing page



You can create an interactive report in three different ways when you create the following:

- A new database application by using the **Create App Wizard** (screenshot 1)
- A new page in an existing database application by using the **Create Page Wizard** (screenshot 2)
- A new region on an existing page in a database application in Page Designer (screenshot 3)

Creating an Interactive Report



Create Interactive Report

Page Attributes

Type: Interactive Report

* Page Number: 6

* Page Name: Projects Master Report

Page Mode: Normal | Modal Dialog

Breadcrumb: Breadcrumb

Parent Entry: No parent entry

Entry Name: Projects Master Report

Create Interactive Report ×

Navigation Menu

Navigation Preference:
○ Do not associate this page with a navigation menu entry
● Create a new navigation menu entry
○ Identify an existing navigation menu entry for this page

* New Navigation Menu Entry: Projects Master Report

Parent Navigation Menu Entry: - No parent selected - / Home / Project Status Report / Project List View

Create Interactive Report

Report Source

Data Source: Local Database | REST Enabled SQL Service | Web Source

* Source Type: ○ Table ● SQL Query

* Enter a SQL SELECT statement

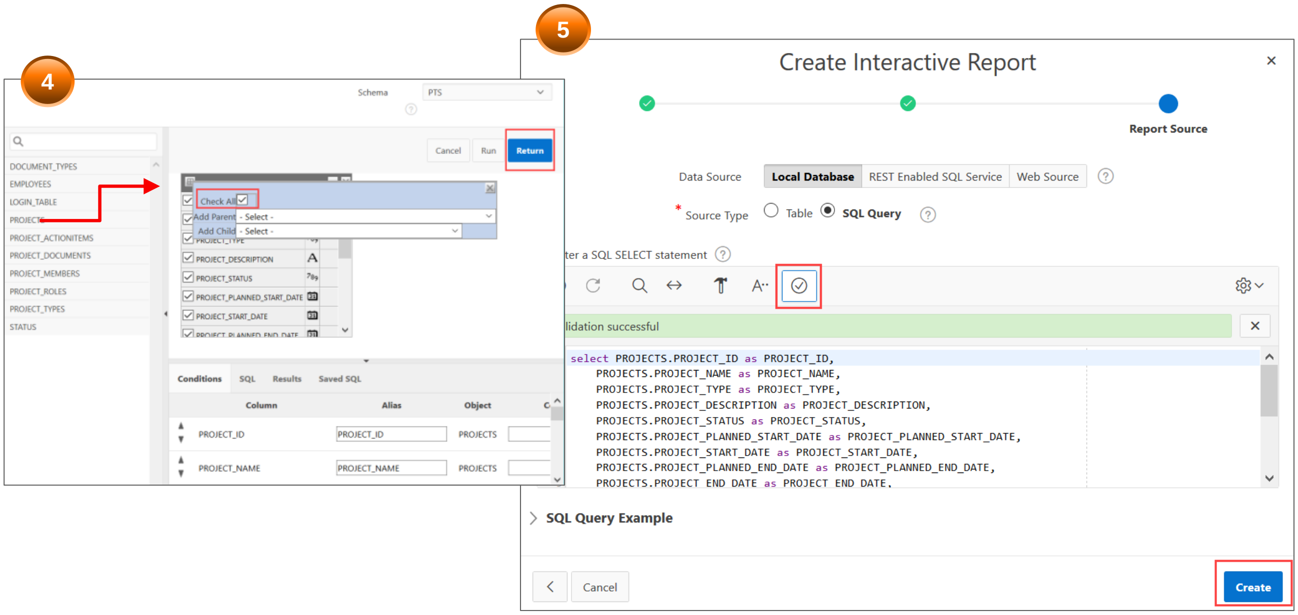
In this lesson, you create an interactive report by using the Create Page Wizard.

Note: You can also create an interactive report with a form. You will learn how to create a form and link it to an interactive report in the lesson titled “Managing Forms.”

To create an interactive report:

1. Sign in to Oracle APEX as a developer and click **App Builder** on the home page.
2. Click the **Project Tracking System** icon and click **Create Page.**
3. Select **Report** and then select **Interactive Report**
4. Enter the following values and retain the default values for other fields (screenshot 2):
   - **Page Name:** `Projects Master Report`
   - **Breadcrumbs:** Select **Breadcrumbs**
   - **Parent Entry:** `-No parent entry-`
   - **Entry Name:** `Project Master Report` (selected by default)
   - Click **Next**.
5. For the Navigation Menu:
   - **Navigation Preference**: `Create a new navigation menu entry.`
   - **New Navigation Menu Entry:** `Project Master Report`
   - **Parent Navigation Menu Entry:** `- no parent selected-`
   - Click **Next**.
6. For Data Source, click **Local Database**; for Source Type, click **SQL Query** and then click the **Query Builder** icon (screenshot 3).

Creating an Interactive Report



Check All ☑
Add Parent: - Select -
Add Child: - Select -

☑ PROJECT_TYPE
☑ PROJECT_DESCRIPTION
☑ PROJECT_STATUS
☑ PROJECT_PLANNED_START_DATE
☑ PROJECT_START_DATE
☑ PROJECT_PLANNED_END_DATE

Column	Alias	Object
PROJECT_ID	PROJECT_ID	PROJECTS
PROJECT_NAME	PROJECT_NAME	PROJECTS

Data Source: Local Database

* Source Type: ○ Table ● SQL Query

ter a SQL SELECT statement

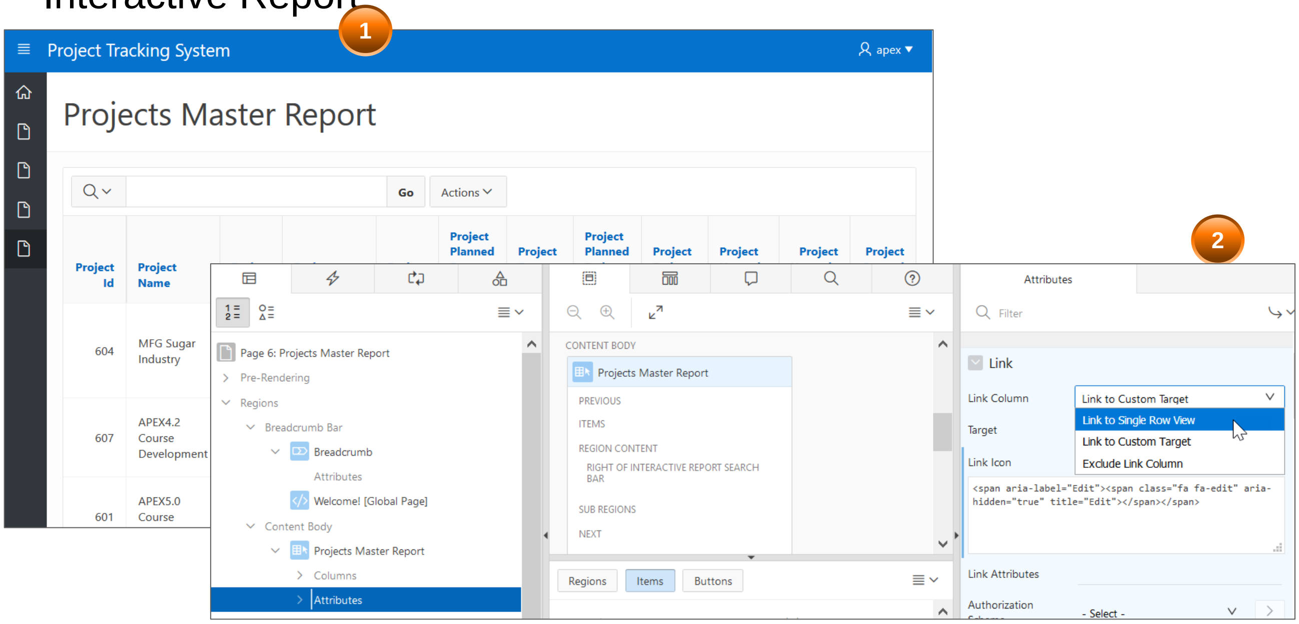
lidation successful

```
select PROJECTS.PROJECT_ID as PROJECT_ID,
    PROJECTS.PROJECT_NAME as PROJECT_NAME,
    PROJECTS.PROJECT_TYPE as PROJECT_TYPE,
    PROJECTS.PROJECT_DESCRIPTION as PROJECT_DESCRIPTION,
    PROJECTS.PROJECT_STATUS as PROJECT_STATUS,
    PROJECTS.PROJECT_PLANNED_START_DATE as PROJECT_PLANNED_START_DATE,
    PROJECTS.PROJECT_START_DATE as PROJECT_START_DATE,
    PROJECTS.PROJECT_PLANNED_END_DATE as PROJECT_PLANNED_END_DATE,
    PROJECTS.PROJECT_END_DATE as PROJECT_END_DATE,
```

ORACLE

7. In the Source dialog box, select the `PROJECTS` table
8. In the `PROJECTS` table dialog box, select **Check All**. As shown in screenshot 5 in the slide, selecting Check All included all the columns such as `PROJECT.NAME`, `PROJECT.TYPE`, `PROJECT.DESCRIPTION`, `PROJECT.STATUS`, and so on from the `PROJECTS` table in the *Project Master Report* (screenshot 4).
9. Click **Return** (screenshot 4). This takes you back to Report Source dialog box.
10. In the Report Source dialog box in the Create Page Wizard, validate the SQL statement and click **Create** (screenshot 5). You are redirected to Page Designer.
11. Click **Save** and **Run.** This completes the task of creating the Project Master report.

Interactive Report



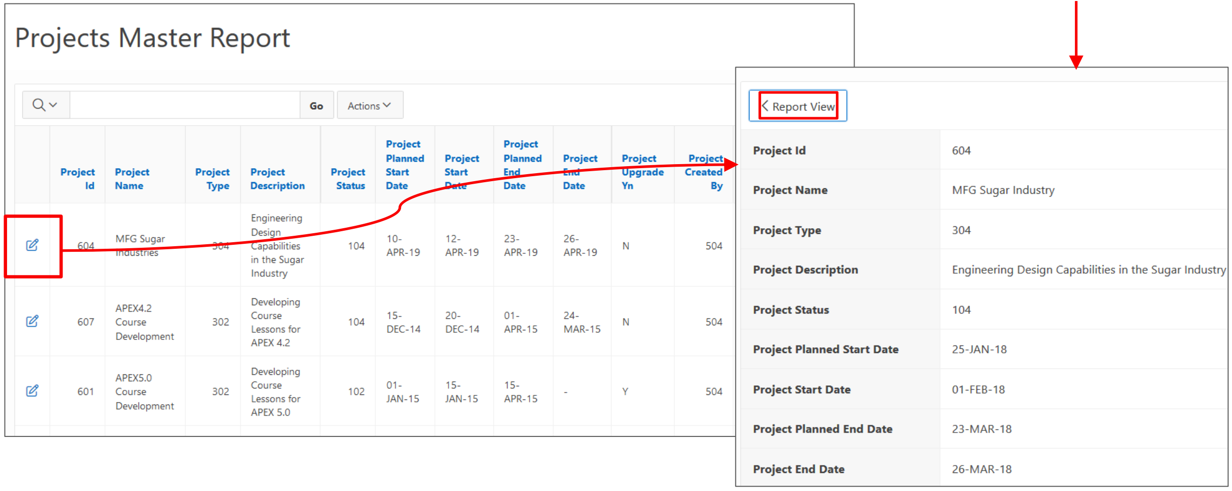
```
<span aria-label="Edit"><span class="fa fa-edit" aria-hidden="true" title="Edit"></span></span>
```

The *Project Master Report* is generated as shown in the slide. Let's modify this report to include a link for each row. When you click the link for a row, the details of the row are displayed in a separate window.

1. On the Developer toolbar, click **Quick Edit.** The report page opens in Page Designer.
2. On the Rendering tab, click **Attributes** under **Project Master Report.**
3. In the right pane, under **Link,** click **Link to Single Row View**. The single row view enables you to view the details of the selected row in a separate window (screenshot 2).
   Note: You also have the option to select **Link to Custom Target**, using which you can link it to a specific page in the same application or to a specific page in another application or link it to a URL also. In this example, you select **Link to Single Row View.**
4. Click **Save** and **Run.**

You will observe that the *Project Master Report* (on the next slide - slide10) now contains the pencil icon for each row, which is the **Link Column.**

Interactive Report with Link to Single Row View



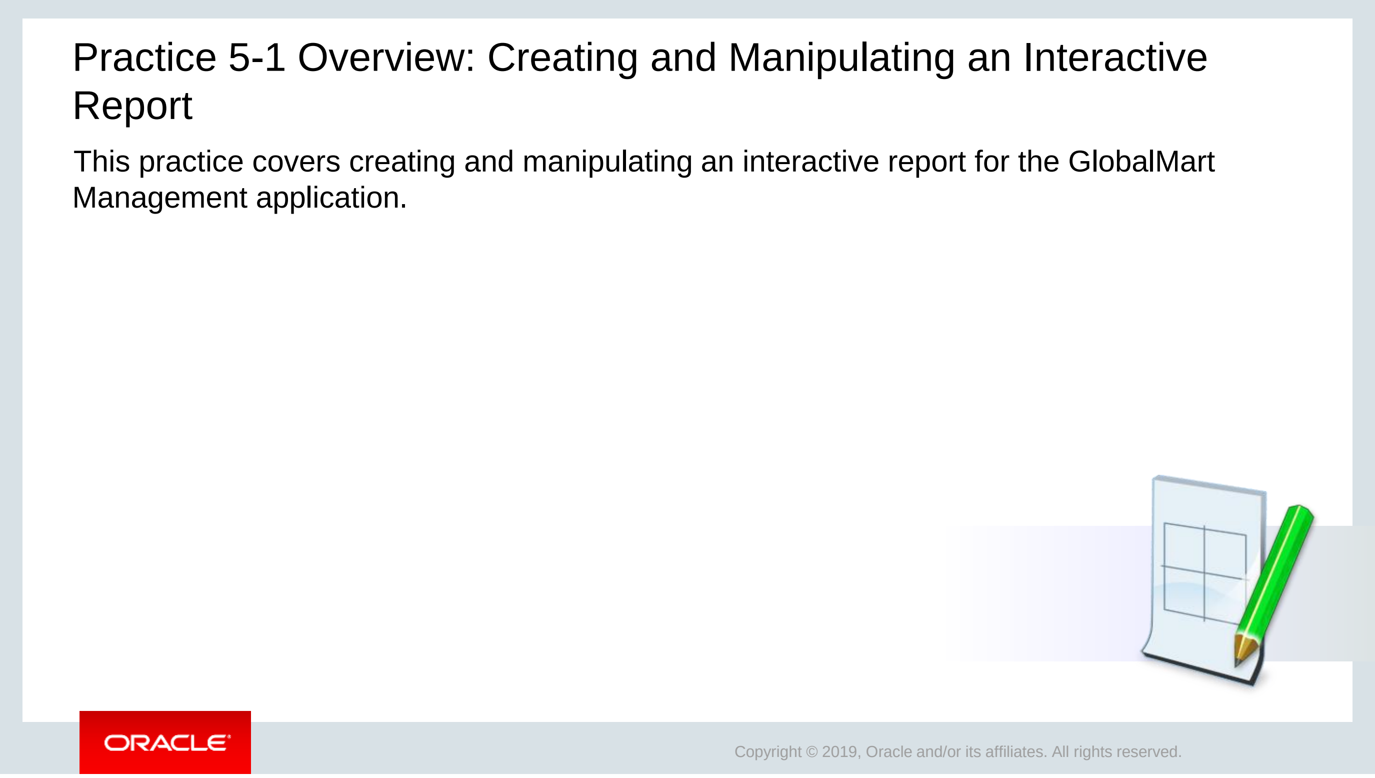
Single Row View of Project ID 604

Projects Master Report

	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Project End Date	Project Upgrade Yn	Project Created By
✎	604	MFG Sugar Industries	304	Engineering Design Capabilities in the Sugar Industry	104	10-APR-19	12-APR-19	23-APR-19	26-APR-19	N	504
✎	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14	01-APR-15	24-MAR-15	N	504
✎	601	APEX5.0 Course Development	302	Developing Course Lessons for APEX 5.0	102	01-JAN-15	15-JAN-15	15-APR-15	-	Y	504

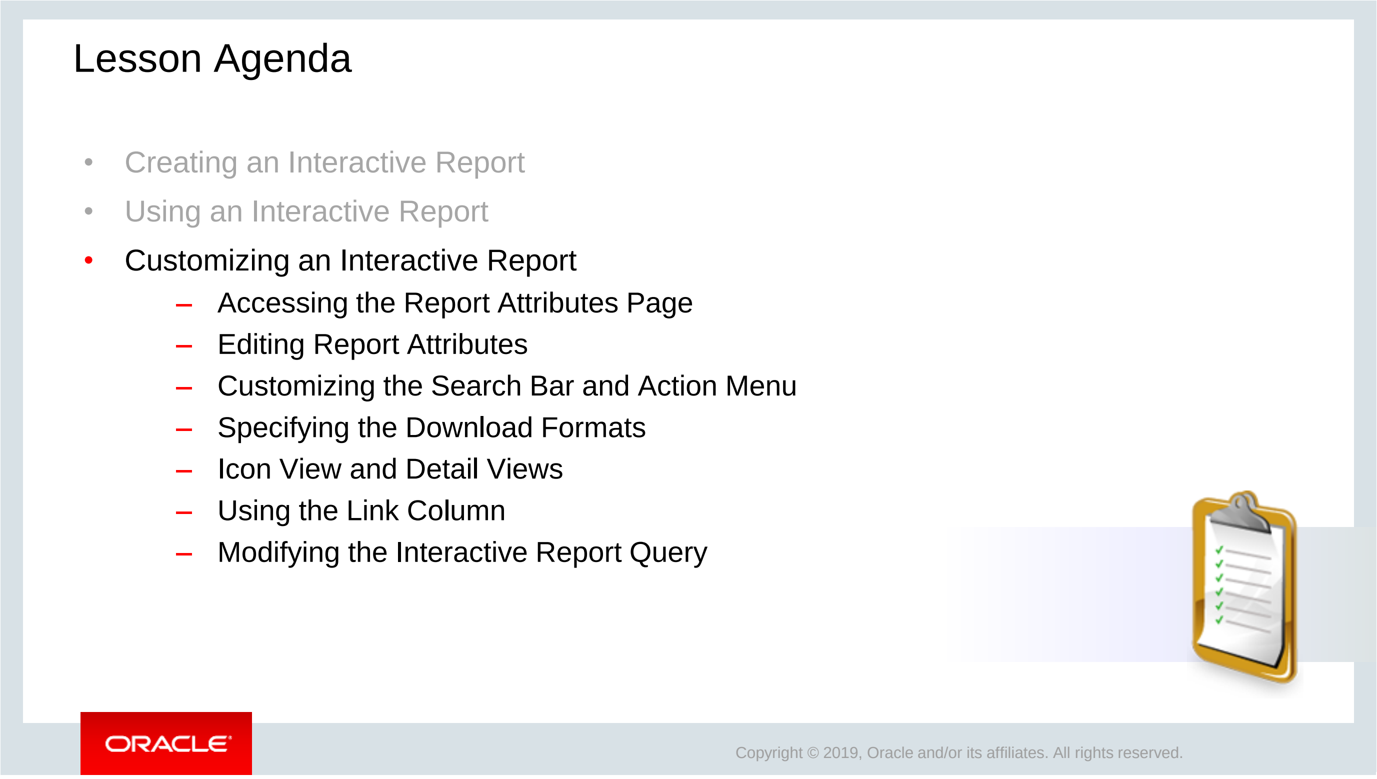
Project Id	604
Project Name	MFG Sugar Industry
Project Type	304
Project Description	Engineering Design Capabilities in the Sugar Industry
Project Status	104
Project Planned Start Date	25-JAN-18
Project Start Date	01-FEB-18
Project Planned End Date	23-MAR-18
Project End Date	26-MAR-18

The *Project Master Report* is now generated with **Link to Single Row View** (Link Column), as shown in the slide. It also shows the details of the first row with Project ID 604 in the single row view. Notice that in the Single Row View window, there is the **Report View** button to go back to the parent page, which is the *Project Master Report.*



Practice 5-1 Overview: Creating and Manipulating an Interactive Report

This practice covers creating and manipulating an interactive report for the GlobalMart Management application.



Lesson Agenda

- Creating an Interactive Report
- Using an Interactive Report
- Customizing an Interactive Report
  - Accessing the Report Attributes Page
  - Editing Report Attributes
  - Customizing the Search Bar and Action Menu
  - Specifying the Download Formats
  - Icon View and Detail Views
  - Using the Link Column
  - Modifying the Interactive Report Query

Accessing Report Attributes

To access Report Attributes:

1. Access the page definition where the interactive report is created in Page Designer mode.
2. In Rendering Tree, select the region. The region can be identified by the same name as the report. In this example, it is *Projects Master Report.*

The Property Editor displays the region attributes in the right pane. Attributes are organized into functional groups.

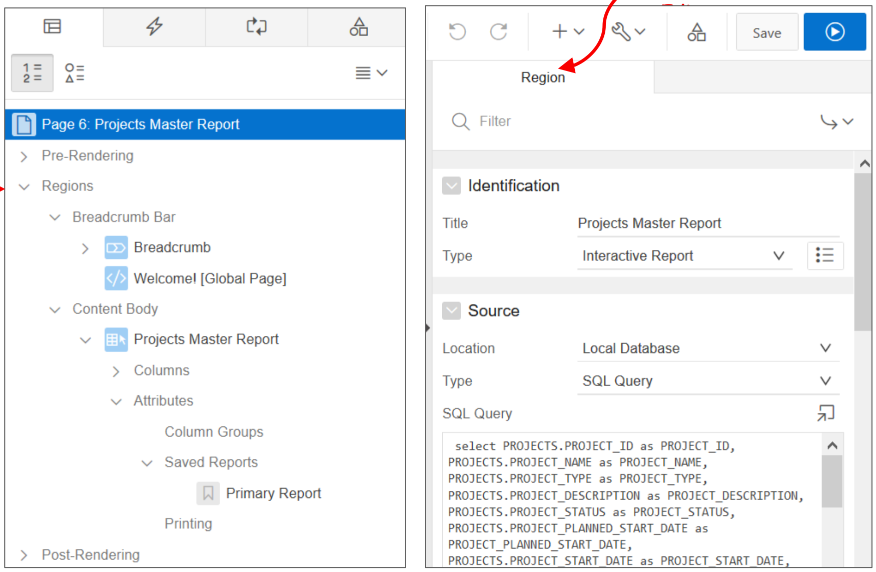
ORACLE

As a developer, you can change the way an interactive report is rendered to users by editing the Report Attributes. The steps to access the **Report Attributes** are shown in the slide.

All the attributes of the report region can be found in the **Property Editor** in the right column. You can find details such as Report Title, Type, Source SQL Query, Report layout, appearance, and other advanced features.

Steve created the *Projects Master Report* as an interactive report to give complete information about all the projects entered into PTS. After creating the report, Steve thought that it will be better to display Project Name instead of Project ID for Project upgrade of field in the report. He also removes Pivot option from the Actions Menu by customizing the *Projects Master Report.*

Editing Report Attributes



Page 6: Projects Master Report
- Pre-Rendering
- Regions
  - Breadcrumb Bar
    - Breadcrumb
    - Welcome! [Global Page]
  - Content Body
    - Projects Master Report
      - Columns
      - Attributes
        - Column Groups
        - Saved Reports
          - Primary Report
        - Printing
- Post-Rendering

Identification

Title: Projects Master Report

Type: Interactive Report

Source

Location: Local Database

Type: SQL Query

SQL Query

```
 select PROJECTS.PROJECT_ID as PROJECT_ID,
PROJECTS.PROJECT_NAME as PROJECT_NAME,
PROJECTS.PROJECT_TYPE as PROJECT_TYPE,
PROJECTS.PROJECT_DESCRIPTION as PROJECT_DESCRIPTION,
PROJECTS.PROJECT_STATUS as PROJECT_STATUS,
PROJECTS.PROJECT_PLANNED_START_DATE as
PROJECT_PLANNED_START_DATE,
PROJECTS.PROJECT_START_DATE as PROJECT_START_DATE,
```

Report Region
Attributes in Property

Report Region

Collapsible tabs under Region

When you open the page in Page Designer mode and select the report region in "Page Rendering," all the report attributes will be listed in the Property Editor in the right column of the page. All the attributes are organized into functional groups.

To edit any report attribute, locate it in the Property Editor and update its value as per the change required. The Messages tab will highlight errors in red color if any of the values updated in the Properties Pane is invalid.

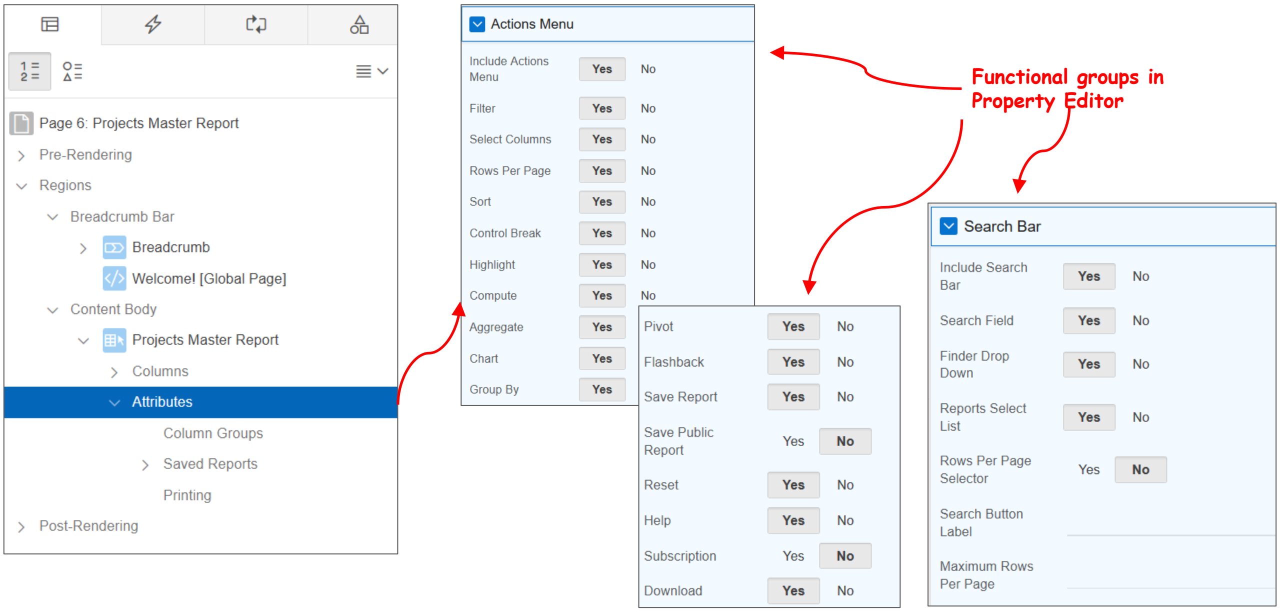
Update the report attributes as required and click the "Save and Run Page" icon on the page to see the updated report.

All the report attributes are not associated only with the Report Region directly. Some of the attributes are organized under collapsible tabs shown under Report Region.

All the columns selected from the database to show on the report are listed under the Columns tab. By clicking each column, its properties appear in the Properties Pane on the right column. This allows you to update the properties at an individual column level. For example, by changing the "Type" to "Hidden", you can make any column not to be listed even in the "Do Not Display" box of the report.

The Attributes tab allows developers to deal with the properties related to Column Groups, Saved Reports, and Printing.

Customizing the Search Bar and Actions Menu



Functional groups in Property Editor

☑ Actions Menu

Include Actions Menu	[✓ Yes] No
Filter	[✓ Yes] No
Select Columns	[✓ Yes] No
Rows Per Page	[✓ Yes] No
Sort	[✓ Yes] No
Control Break	[✓ Yes] No
Highlight	[✓ Yes] No
Compute	[✓ Yes] No
Aggregate	[✓ Yes]
Chart	[✓ Yes]
Group By	[✓ Yes]
Pivot	[✓ Yes] No
Flashback	[✓ Yes] No
Save Report	[✓ Yes] No
Save Public Report	Yes [✓ No]
Reset	[✓ Yes] No
Help	[✓ Yes] No
Subscription	Yes [✓ No]
Download	[✓ Yes] No

☑ Search Bar

Include Search Bar	[✓ Yes] No
Search Field	[✓ Yes] No
Finder Drop Down	[✓ Yes] No
Reports Select List	[✓ Yes] No
Rows Per Page Selector	Yes [✓ No]
Search Button Label	
Maximum Rows Per Page	

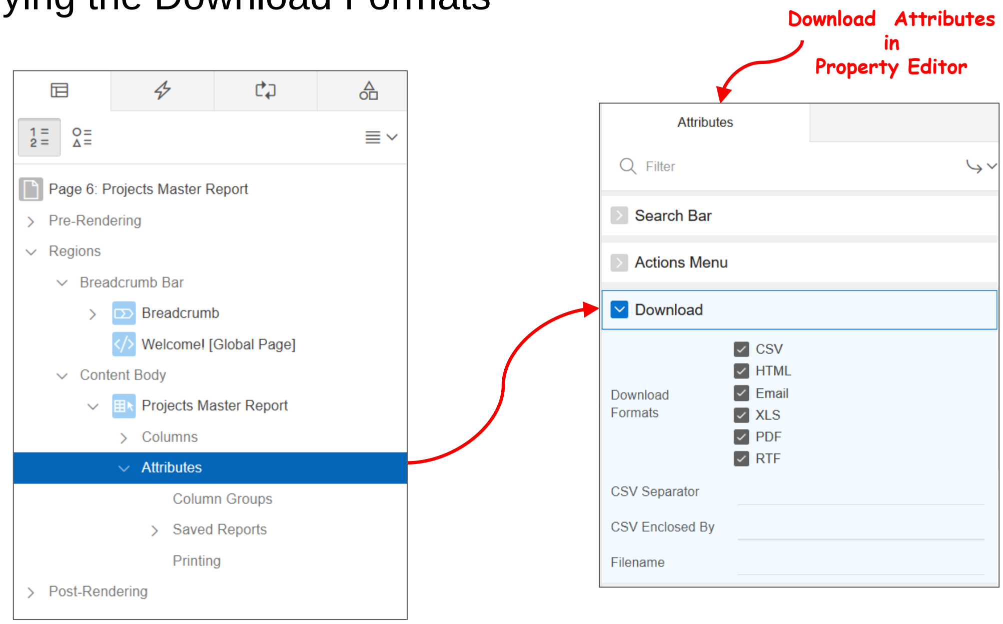
You can choose whether or not to include a search bar in an interactive report. By default, a search bar is included in an interactive report. If you set Include Search Bar to No, the search bar and all its components are removed from the interactive report. You can specify which components of the search bar should be displayed.

You can also control the options that are displayed under the Actions menu. All the actions are selected by default. Deselect the option that you do not want in the Actions menu of the report.

Click **Save and Run Page** to save the changes that you made to the report attributes.

Steve removes the Pivot option from the Actions Menu by accessing the Actions Menu Properties from the Property Editor.

Specifying the Download Formats



Download Attributes in Property Editor

☑ Download

Download Formats: ☑ CSV ☑ HTML ☑ Email ☑ XLS ☑ PDF ☑ RTF

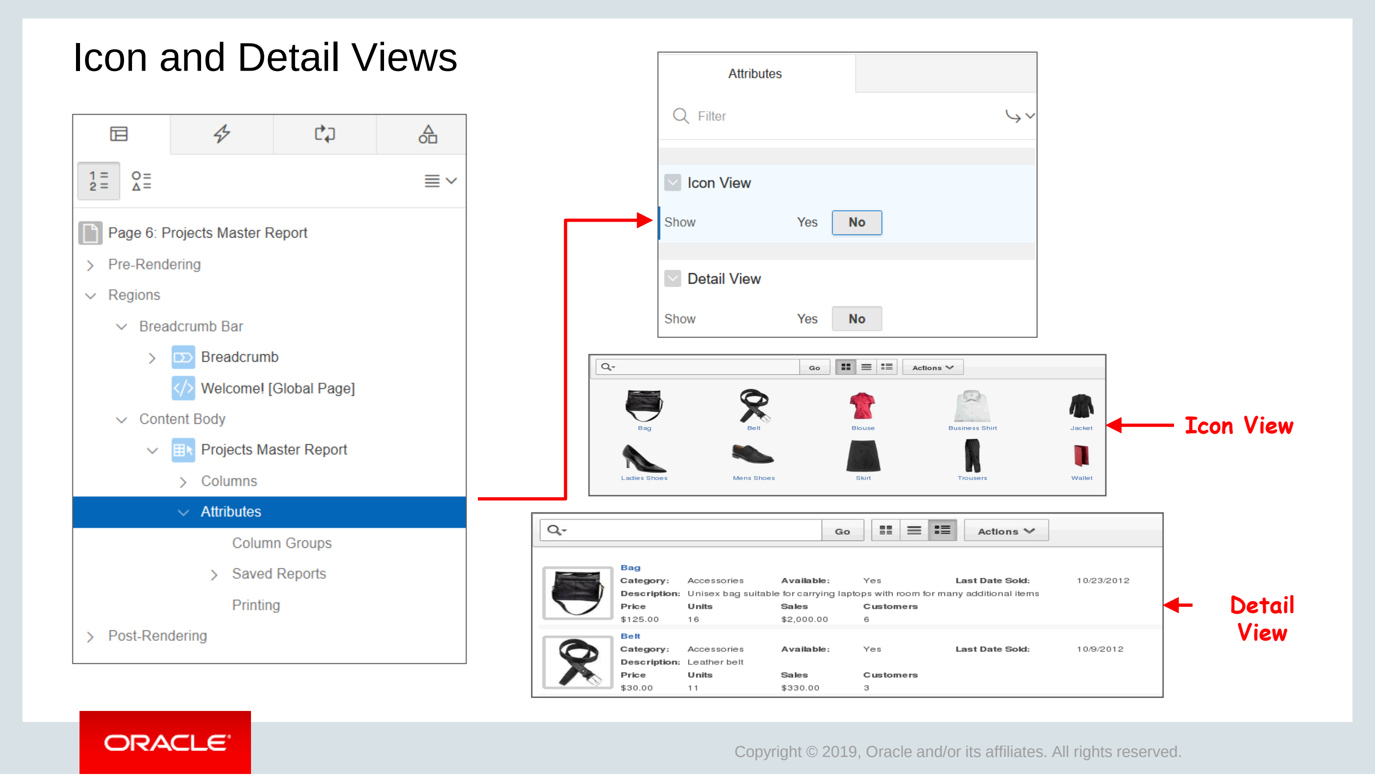
CSV Separator

CSV Enclosed By

Filename

On the Download Attributes page in the Property Editor, you can specify the formats in which users can download the report data. The available formats are CSV, HTML, Email, XLS, PDF, and RTF.

Click **Save and Run Page** to save the changes that you made to the report attributes.



Icon and Detail Views

Icon View

Show Yes [✓ No]

Detail View

Show Yes [✓ No]

Icon View

Bag
Category: Accessories Available: Yes Last Date Sold: 10/23/2012
Description: Unisex bag suitable for carrying laptops with room for many additional items
Price Units Sales Customers
$125.00 16 $2,000.00 6

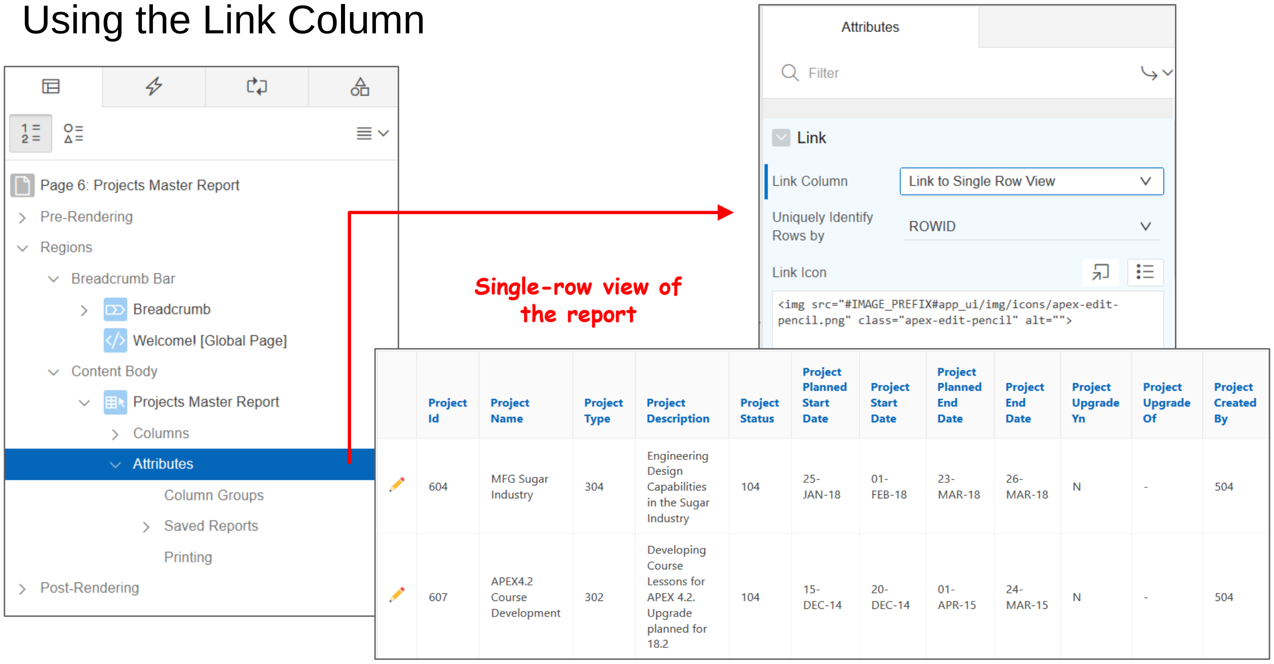
Belt
Category: Accessories Available: Yes Last Date Sold: 10/9/2012
Description: Leather belt
Price Units Sales Customers
$30.00 11 $330.00 3

Detail View

On the Icon View and Detail View tabs, you can define Icon and Detail views for an interactive report. When you enable each of these views, an icon is created on the search bar of the interactive report.

The Icon View is ideal when you have an image column in your report. The Detail View enables you to display the report data by using HTML formatting. Examples of these views (shown in the screenshot in the slide) are included on the Products tab in the Sample Database application that is installed in each Application Express workspace by default.

Using the Link Column



Link Column: Link to Single Row View

Uniquely Identify Rows by: ROWID

Link Icon

```
<img src="#IMAGE_PREFIX#app_ui/img/icons/apex-edit-pencil.png" class="apex-edit-pencil" alt="">
```

Single-row view of the report

	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Project End Date	Project Upgrade Yn	Project Upgrade Of	Project Created By
✎	604	MFG Sugar Industry	304	Engineering Design Capabilities in the Sugar Industry	104	25-JAN-18	01-FEB-18	23-MAR-18	26-MAR-18	N	-	504
✎	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2. Upgrade planned for 18.2	104	15-DEC-14	20-DEC-14	01-APR-15	24-MAR-15	N	-	504

For an interactive report, you can specify a link column. You can create a column link to a single-row view or to another page in the application.

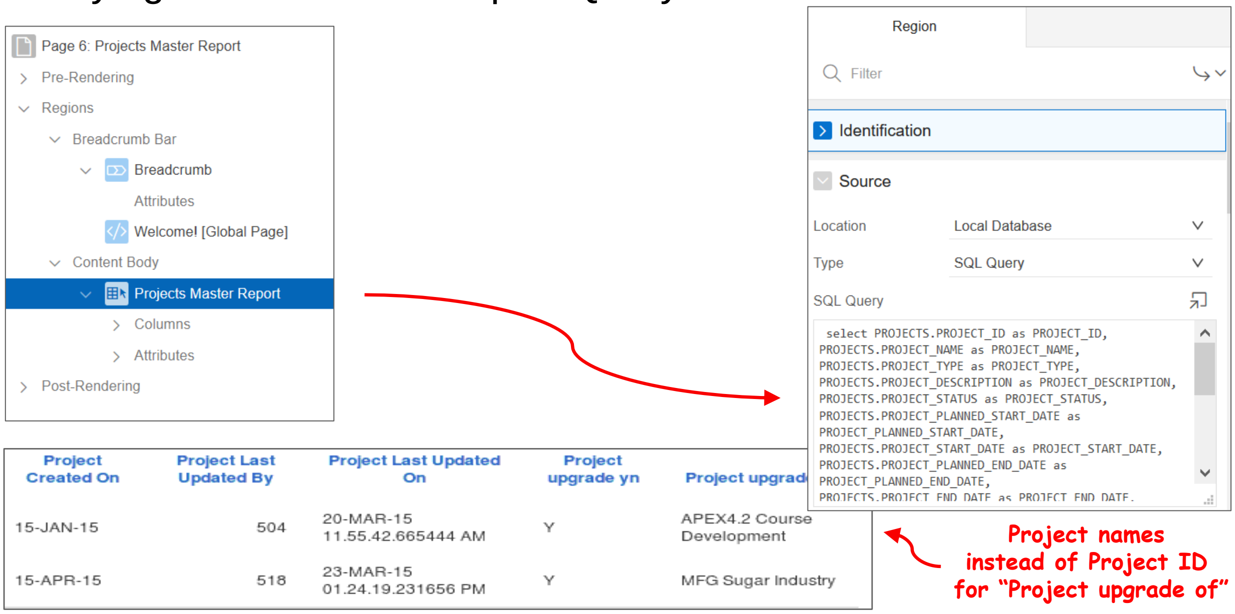
The single-row view is used by default when you create an interactive report. The single-row view is a display-only view of all the columns in the report. If you have a column in your query but it is hidden in Column Attributes, it will not be displayed in the single-row view. If you have a column that you have hidden in the report by using Select Columns in the Actions menu, it will appear in the single-row view. From the single-row view, you can navigate through all the rows by clicking the Previous and Next buttons. To return to the report, you can click the Review View button.

If you choose to link to a custom page, you can pass item session state values. Linking to a custom page is explained in detail in the lesson titled “Managing Forms.”

You can also completely remove the link column from the report. A link column cannot be sorted, hidden, or moved by an end user.

Click **Save and Run Page** to save the changes that you made to the report attributes.

Modifying the Interactive Report Query



Location: Local Database

Type: SQL Query

SQL Query

```
 select PROJECTS.PROJECT_ID as PROJECT_ID,
PROJECTS.PROJECT_NAME as PROJECT_NAME,
PROJECTS.PROJECT_TYPE as PROJECT_TYPE,
PROJECTS.PROJECT_DESCRIPTION as PROJECT_DESCRIPTION,
PROJECTS.PROJECT_STATUS as PROJECT_STATUS,
PROJECTS.PROJECT_PLANNED_START_DATE as
PROJECT_PLANNED_START_DATE,
PROJECTS.PROJECT_START_DATE as PROJECT_START_DATE,
PROJECTS.PROJECT_PLANNED_END_DATE as
PROJECT_PLANNED_END_DATE,
PROJECTS.PROJECT_END_DATE as PROJECT_END_DATE,
```

Project Created On	Project Last Updated By	Project Last Updated On	Project upgrade yn	Project upgrad
15-JAN-15	504	20-MAR-15 11.55.42.665444 AM	Y	APEX4.2 Course Development
15-APR-15	518	23-MAR-15 01.24.19.231656 PM	Y	MFG Sugar Industry

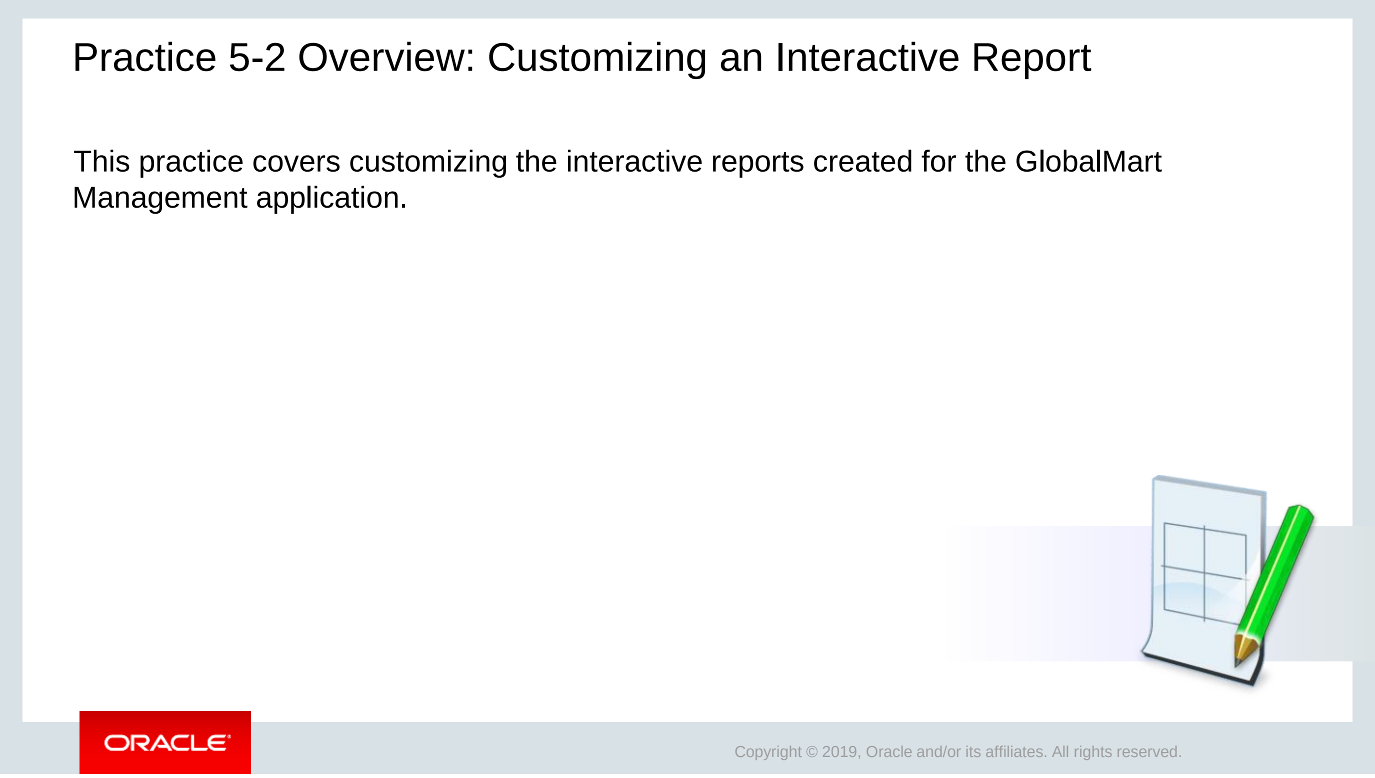
Project names instead of Project ID for "Project upgrade of"

You can change the query that is executed when the report is run. To do this, perform the following steps:

1. From the page definition in Page Designer mode, locate and select Report Region under Page Rendering.
2. In the Property Editor, the properties of the report region can be seen.
3. Locate **SQL Query** under **Source** and replace the old query with a new query.
4. Alternatively, you can rebuild the new query using a Query Builder.
5. Click **Save and Run Page.**
6. In the confirmation window, click **Apply Changes**.

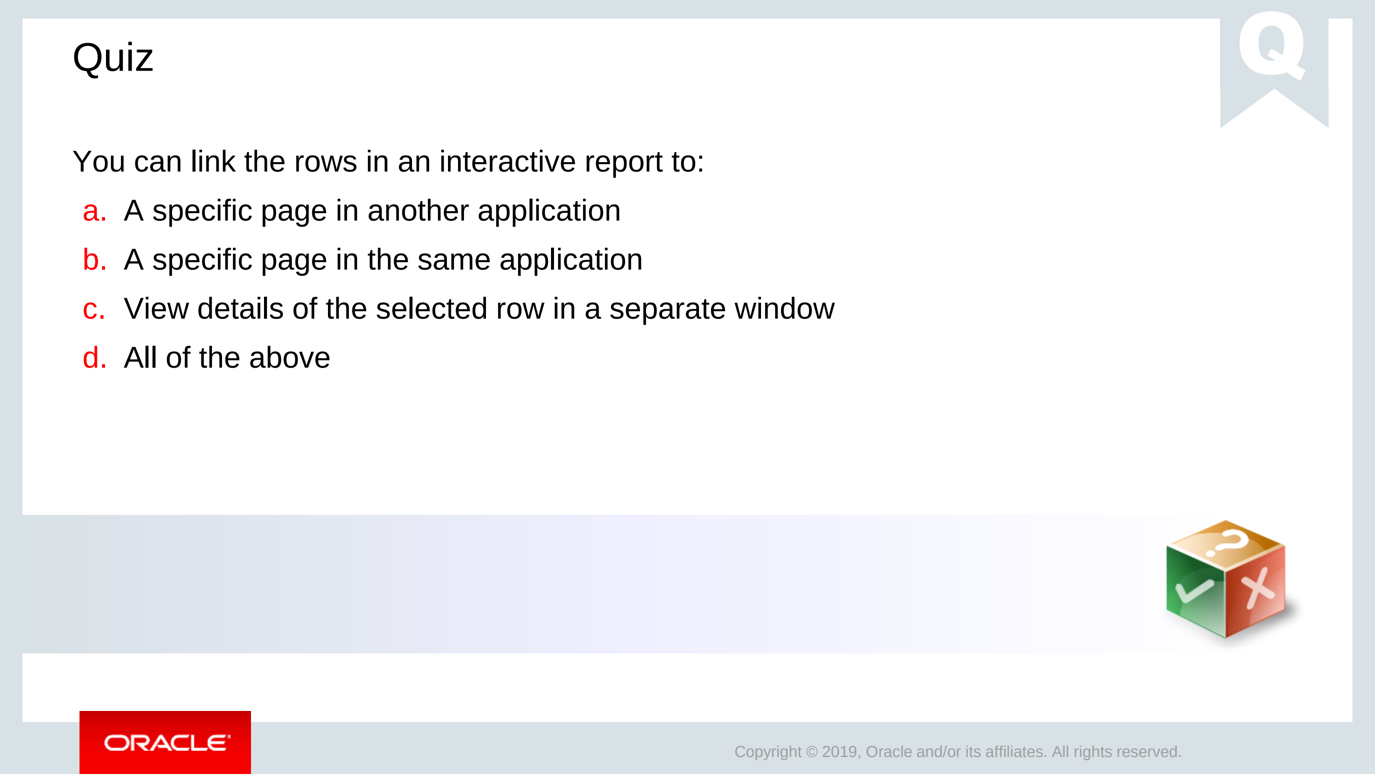
If you add columns to the query, they are not displayed when the report is run. In this case, to see the changes in your report, you must reset the report.

To replace Project ID with the Project Name for the "Project upgrade of" field, Steve created a new query using Join. The SQL query used for creating the Projects Master Report is replaced with the new query.



Practice 5-2 Overview: Customizing an Interactive Report

This practice covers customizing the interactive reports created for the GlobalMart Management application.

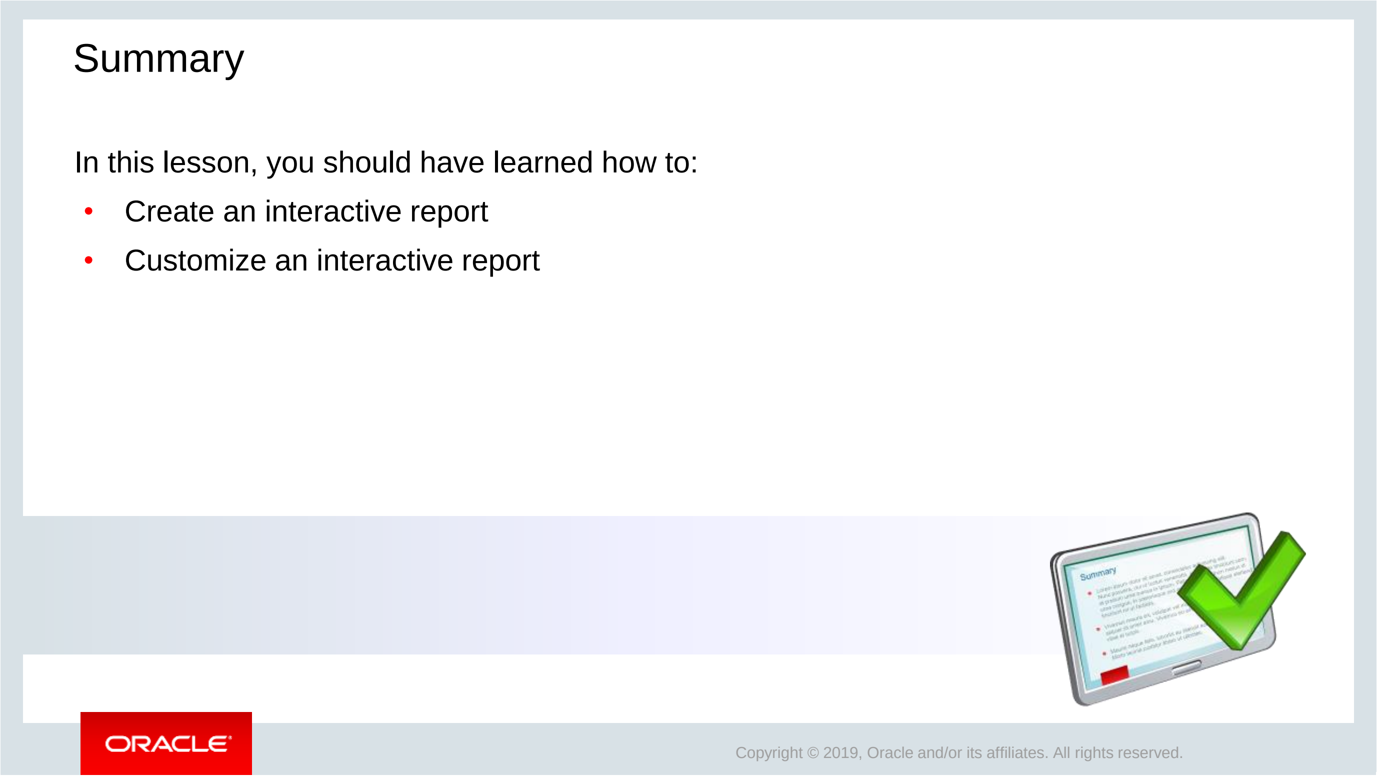


Quiz

You can link the rows in an interactive report to:

a. A specific page in another application
b. A specific page in the same application
c. View details of the selected row in a separate window
d. All of the above

Answer: d



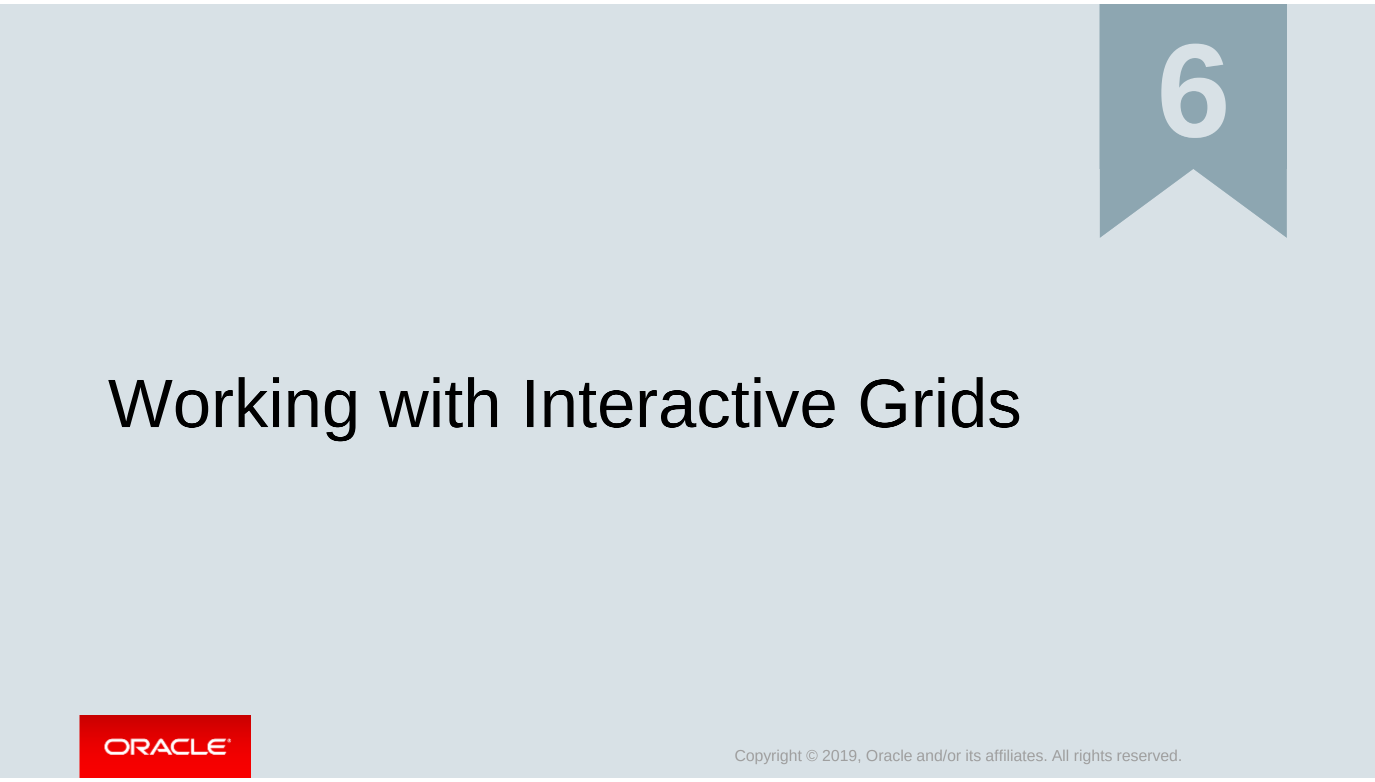
Summary

In this lesson, you should have learned how to:

- Create an interactive report
- Customize an interactive report

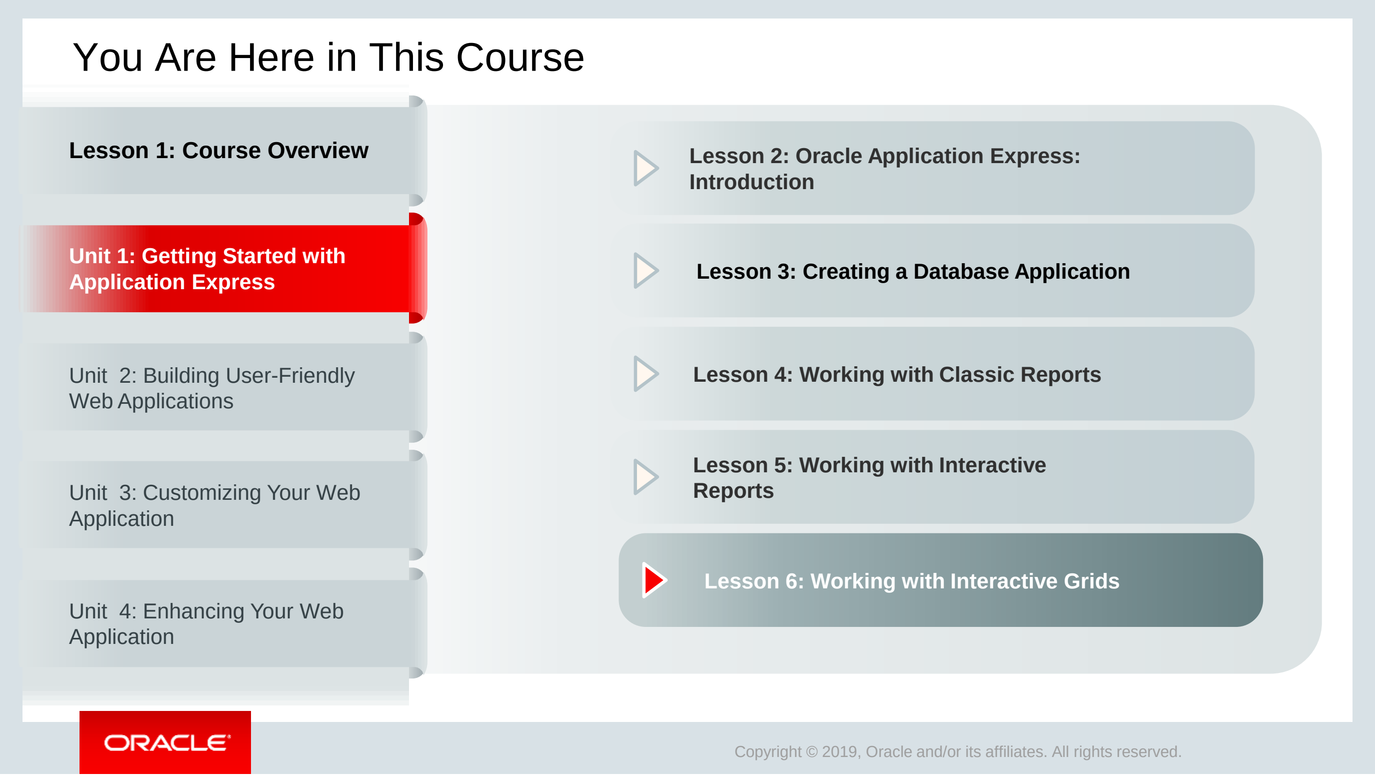
ORACLE®

In this lesson, you learned how to create and use an interactive report.



6

Working with Interactive Grids



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

▷ **Lesson 2: Oracle Application Express: Introduction**

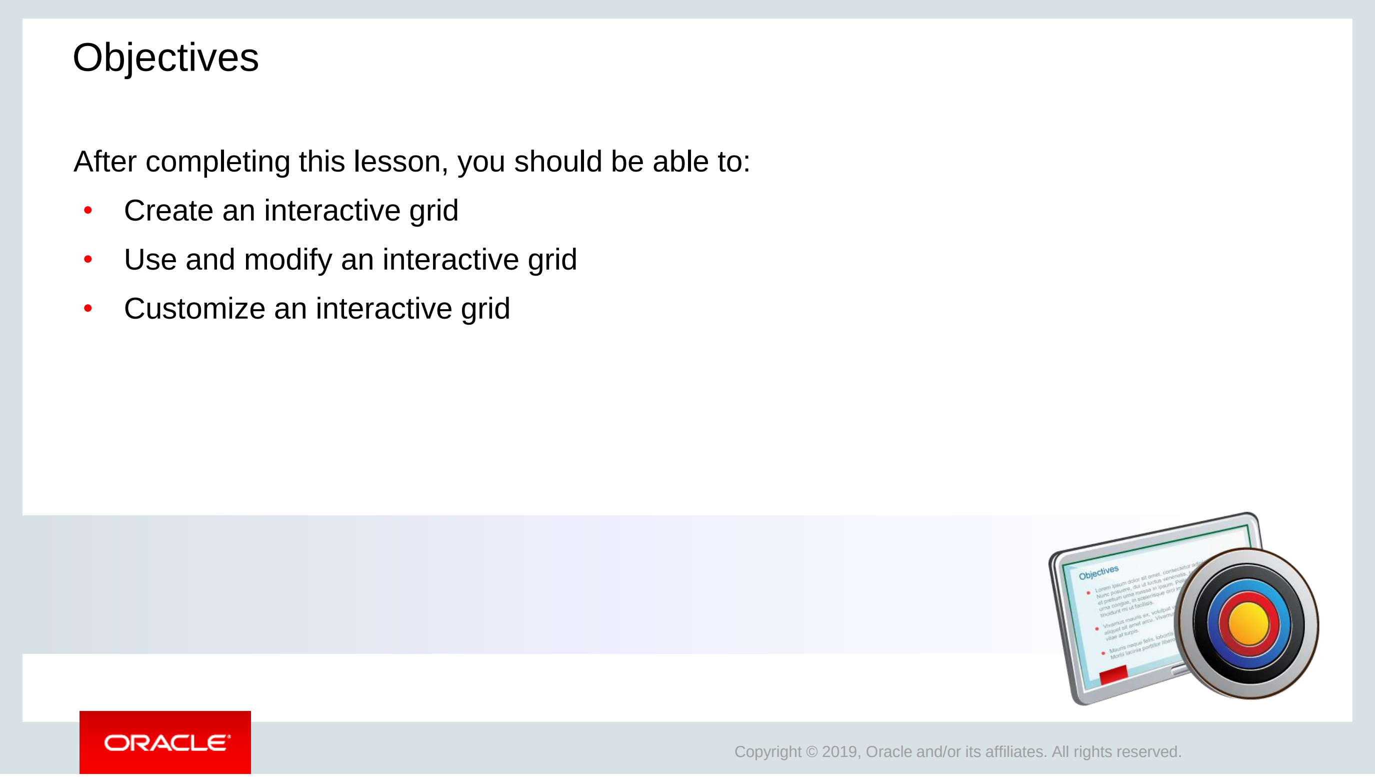
▷ **Lesson 3: Creating a Database Application**

▷ **Lesson 4: Working with Classic Reports**

▷ **Lesson 5: Working with Interactive Reports**

▶ **Lesson 6: Working with Interactive Grids**

This slide shows a graphical representation of the entire course highlighting the lesson, which is dealt with in these slides.



Objectives

After completing this lesson, you should be able to:

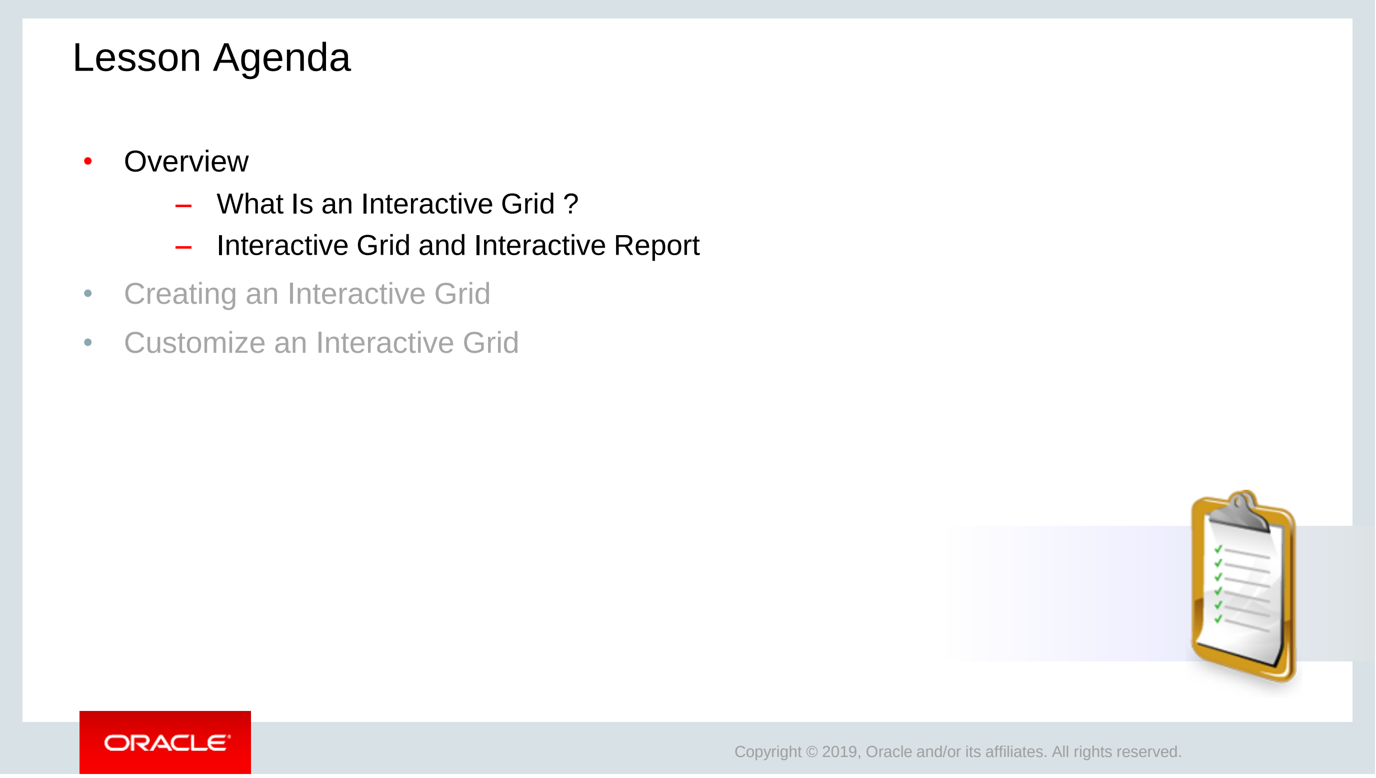
- Create an interactive grid
- Use and modify an interactive grid
- Customize an interactive grid

This lesson introduces you to interactive grids in Oracle Application Express. In this lesson, you learn about interactive grids and how to create and use them.

Steve Explores the Interactive Grid

After having successfully created project status reports for the Project Tracking System (PTS) application, Steve has realized that while managing projects, there is another requirement to manage and track the different project documents. Some of the key documents that require close tracking and management are project plan documents, SQL scripts related to a project, schema-related documents, data modelling diagrams, and project deliverable docs. These documents must be accessible to project managers and team members for updates. Additionally, the project manager needs to download the statuses of these documents as a report.

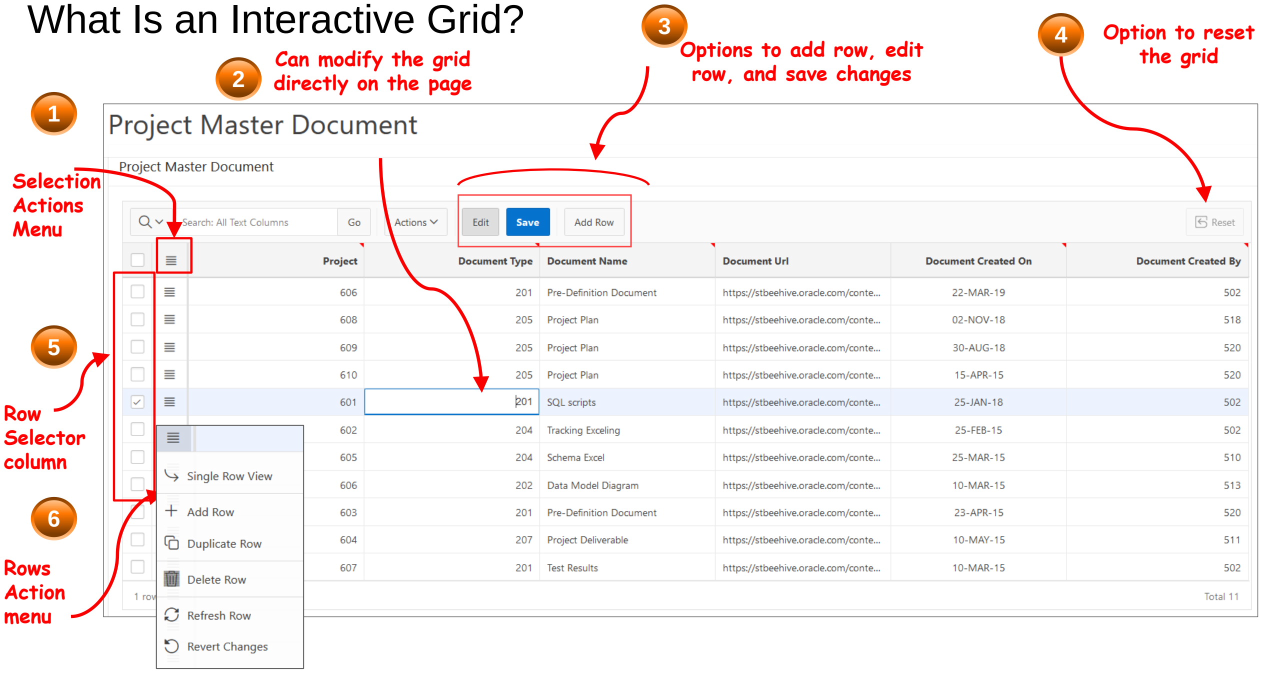
In this scenario, an interactive grid is the best solution. Let's see how Steve manages to create and customize an interactive grid for this requirement.



Lesson Agenda

- Overview
  - What Is an Interactive Grid ?
  - Interactive Grid and Interactive Report
- Creating an Interactive Grid
- Customize an Interactive Grid

What Is an Interactive Grid?



1 Selection Actions Menu

2 Can modify the grid directly on the page

3 Options to add row, edit row, and save changes

4 Option to reset the grid

5 Row Selector column

6 Rows Action menu

Project Master Document

Project Master Document

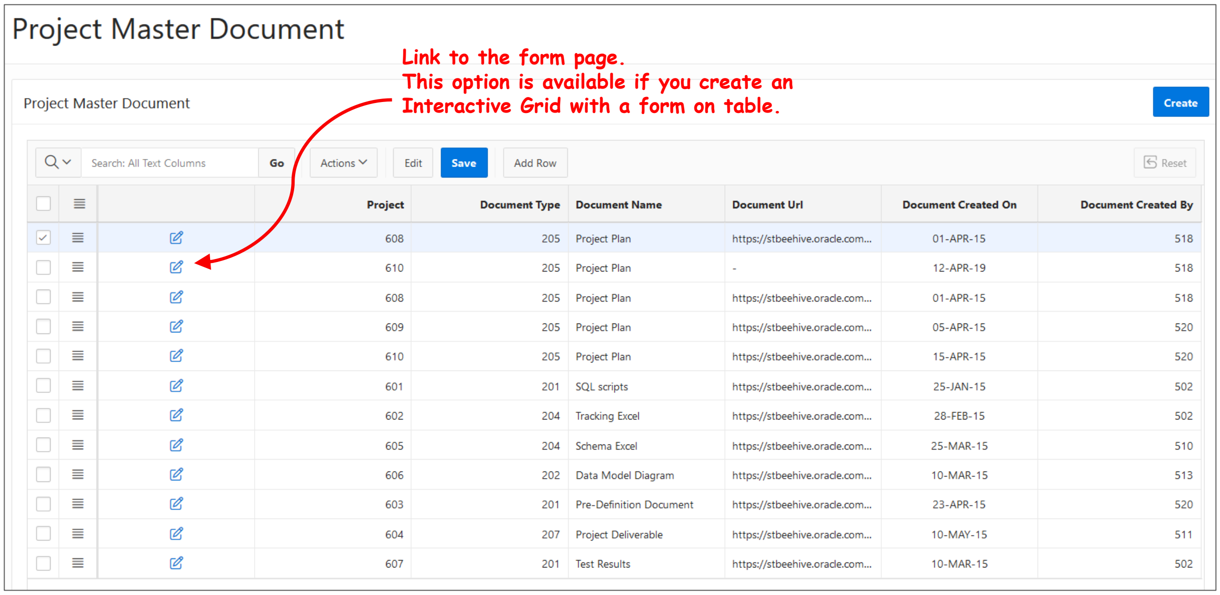
□	Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
□	606	201	Pre-Definition Document	https://stbeehive.oracle.com/conte...	22-MAR-19	502
□	608	205	Project Plan	https://stbeehive.oracle.com/conte...	02-NOV-18	518
□	609	205	Project Plan	https://stbeehive.oracle.com/conte...	30-AUG-18	520
□	610	205	Project Plan	https://stbeehive.oracle.com/conte...	15-APR-15	520
☑	601	201	SQL scripts	https://stbeehive.oracle.com/conte...	25-JAN-18	502
□	602	204	Tracking Exceling	https://stbeehive.oracle.com/conte...	25-FEB-15	502
□	605	204	Schema Excel	https://stbeehive.oracle.com/conte...	25-MAR-15	510
□	606	202	Data Model Diagram	https://stbeehive.oracle.com/conte...	10-MAR-15	513
□	603	201	Pre-Definition Document	https://stbeehive.oracle.com/conte...	23-APR-15	520
□	604	207	Project Deliverable	https://stbeehive.oracle.com/conte...	10-MAY-15	511
□	607	201	Test Results	https://stbeehive.oracle.com/conte...	10-MAR-15	502

1 row Total 11

Like an interactive report, an interactive grid is also a formatted result of a SQL query. But unlike an interactive report, an interactive grid provides some additional customization capabilities, such as:

- Options to add rows, edit, and save changes directly on the page
- Option to reset the interactive grid
- Actions menu for individual rows and for all rows
- Option to modify a row directly

An Interactive Grid with Form



Project Master Document

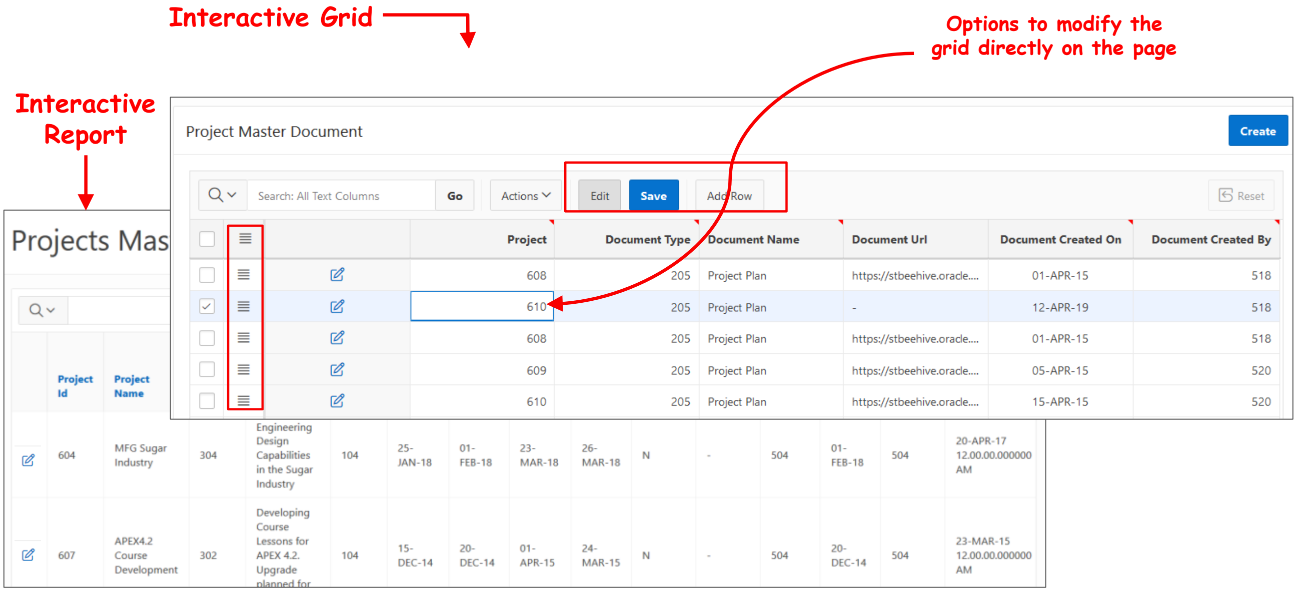
Project Master Document

Link to the form page.
This option is available if you create an Interactive Grid with a form on table.

☐	≡		Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
☑	≡	✎	608	205	Project Plan	https://stbeehive.oracle.com...	01-APR-15	518
☐	≡	✎	610	205	Project Plan	-	12-APR-19	518
☐	≡	✎	608	205	Project Plan	https://stbeehive.oracle.com...	01-APR-15	518
☐	≡	✎	609	205	Project Plan	https://stbeehive.oracle.com...	05-APR-15	520
☐	≡	✎	610	205	Project Plan	https://stbeehive.oracle.com...	15-APR-15	520
☐	≡	✎	601	201	SQL scripts	https://stbeehive.oracle.com...	25-JAN-15	502
☐	≡	✎	602	204	Tracking Excel	https://stbeehive.oracle.com...	28-FEB-15	502
☐	≡	✎	605	204	Schema Excel	https://stbeehive.oracle.com...	25-MAR-15	510
☐	≡	✎	606	202	Data Model Diagram	https://stbeehive.oracle.com...	10-MAR-15	513
☐	≡	✎	603	201	Pre-Definition Document	https://stbeehive.oracle.com...	23-APR-15	520
☐	≡	✎	604	207	Project Deliverable	https://stbeehive.oracle.com...	10-MAY-15	511
☐	≡	✎	607	201	Test Results	https://stbeehive.oracle.com...	10-MAR-15	502

You also have the option to create an interactive grid with a form. If you create an interactive grid with a form, then you get the option to open and edit the form by clicking the pencil icon on the interactive grid.

Interactive Grid Versus Interactive Report



Interactive Grid

Options to modify the grid directly on the page

Interactive Report

Project Master Document

Create

Search: All Text Columns | Go | Actions | Edit | Save | Add Row | Reset

□			Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
□	≡	✎	608	205	Project Plan	https://stbeehive.oracle....	01-APR-15	518
☑	≡	✎	610	205	Project Plan	-	12-APR-19	518
□	≡	✎	608	205	Project Plan	https://stbeehive.oracle....	01-APR-15	518
□	≡	✎	609	205	Project Plan	https://stbeehive.oracle....	05-APR-15	520
□	≡	✎	610	205	Project Plan	https://stbeehive.oracle....	15-APR-15	520

Projects Mas

	Project Id	Project Name														
✎	604	MFG Sugar Industry	304	Engineering Design Capabilities in the Sugar Industry	104	25-JAN-18	01-FEB-18	23-MAR-18	26-MAR-18	N	-	504	01-FEB-18	504	20-APR-17 12.00.00.000000 AM
✎	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2. Upgrade planned for	104	15-DEC-14	20-DEC-14	01-APR-15	24-MAR-15	N	-	504	20-DEC-14	504	23-MAR-15 12.00.00.000000 AM

In the lesson titled “Working with Interactive Reports,” you created the interactive report *Project Master Report*. In this lesson, you will learn how to create an interactive grid with form *Project Master Document.*

Interactive Grid

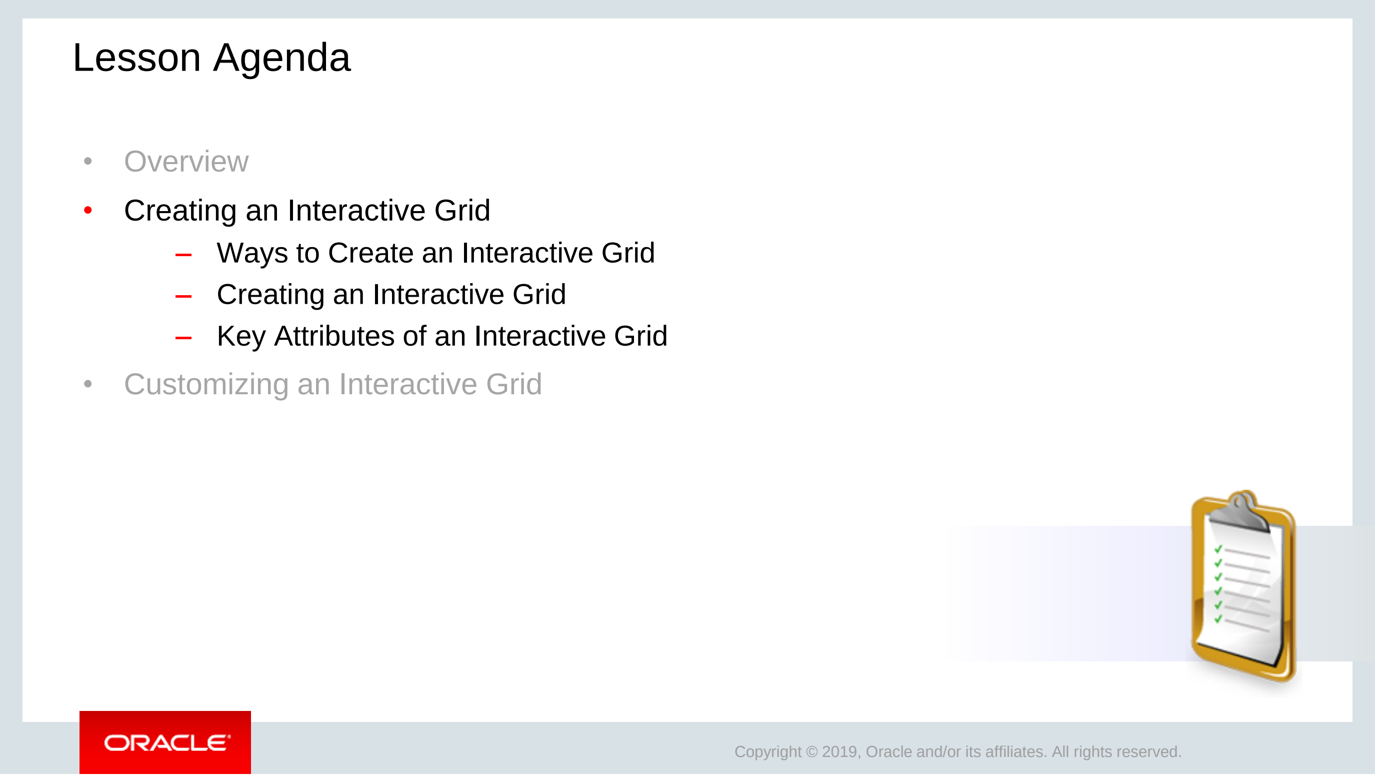
- It is a formatted result of a SQL query.
- It provides a subset of customization options and features that are available in an interactive report.
- It provides additional controls, such as Edit, Save, and Add, using which you can modify data directly on the page.
- An editable interactive grid contains the Row Action menu that allows you to add, edit, duplicate, and refresh rows.
- You can include multiple interactive reports per page.

Interactive Report

- It is a formatted result of a SQL query.
- By default, it includes a search bar, an Actions menu, column heading menus, and Edit icons in the first column of each row.
- You can include multiple interactive reports per page.
- You can restrict the capabilities available to end users, such as disabling download or support for hiding column.
- You can customize how and what data to display when viewing the report.

Classic Report

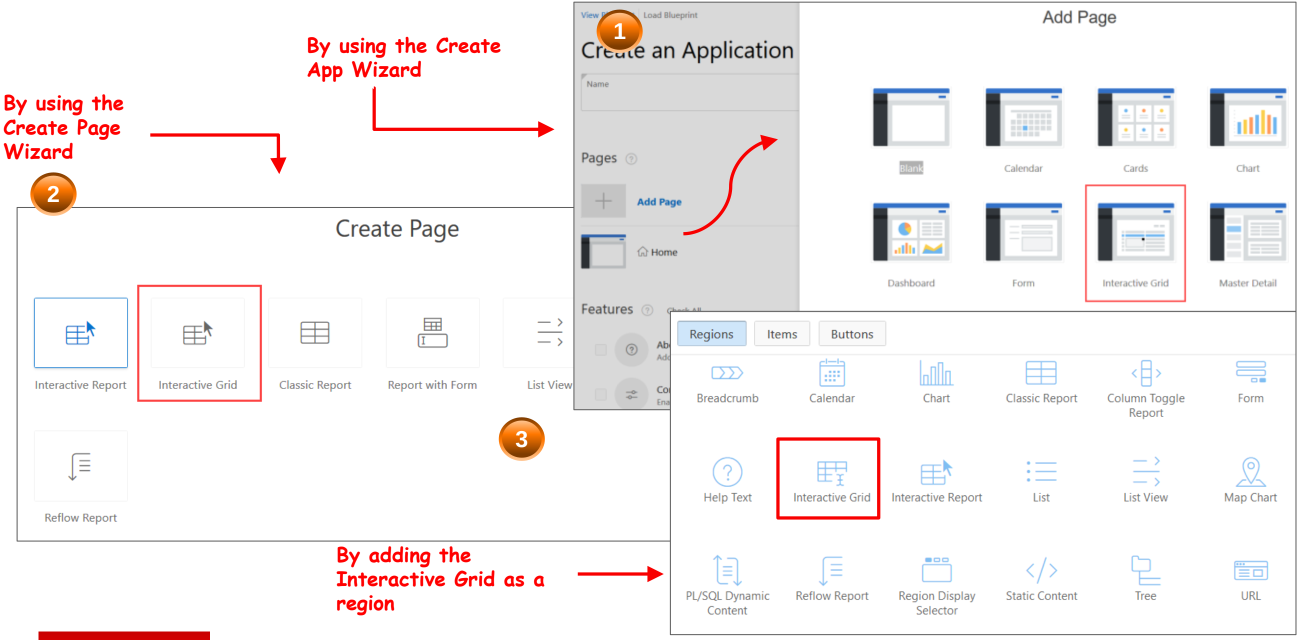
- It is a formatted result of a SQL query.
- No customization is possible.
- It provides sorting and simple filtering.



Lesson Agenda

- Overview
- Creating an Interactive Grid
  - Ways to Create an Interactive Grid
  - Creating an Interactive Grid
  - Key Attributes of an Interactive Grid
- Customizing an Interactive Grid

Ways to Create an Interactive Grid



By using the Create App Wizard

By using the Create Page Wizard

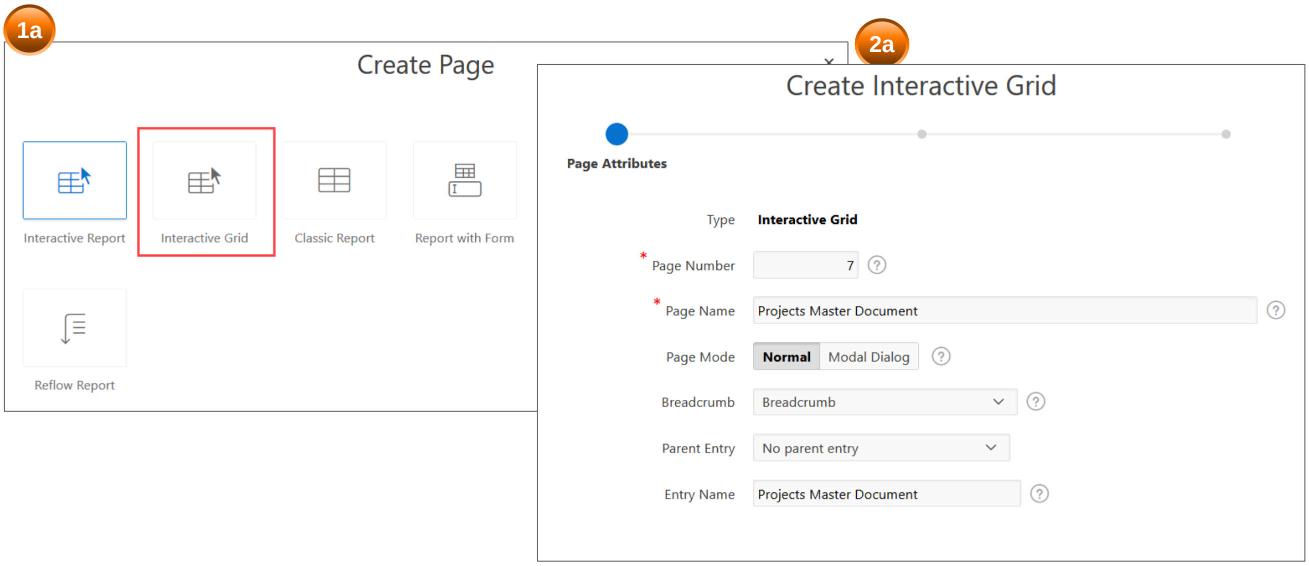
By adding the Interactive Grid as a region

You can create an interactive grid in three ways:

- By using the Create App Wizard to add an Interactive Grid page to the application (screenshot 1)
- By using the Create Page wizard to add an Interactive Grid page to the application (screenshot 2)
- By using the Region under Gallery in Page Designer to drag an interactive grid to add it to the application (screenshot 3)

In this lesson, you learn how to create an interactive grid by using the Create Page Wizard.

Creating an Interactive Grid (without Form)



Create Interactive Grid

Page Attributes

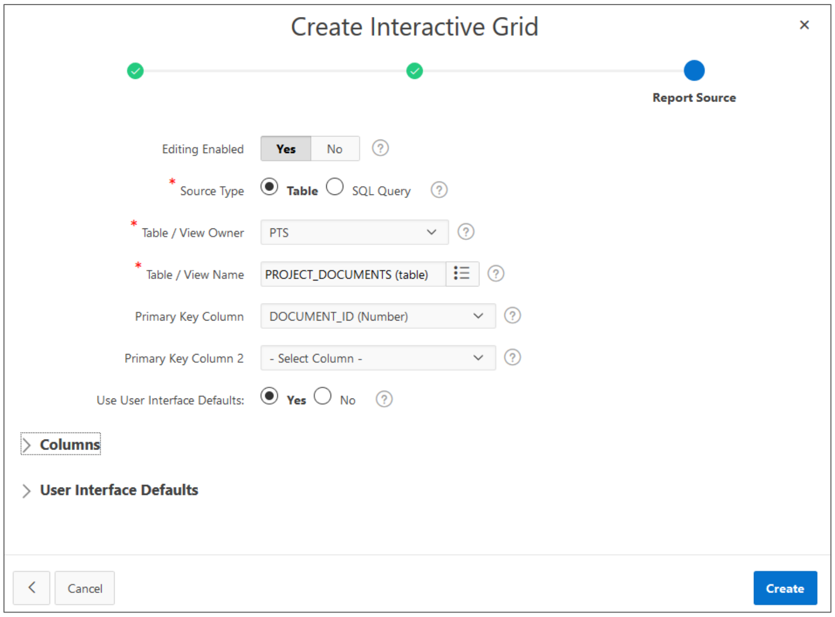
Type	Interactive Grid
* Page Number	7
* Page Name	Projects Master Document
Page Mode	**Normal** Modal Dialog
Breadcrumb	Breadcrumb
Parent Entry	No parent entry
Entry Name	Projects Master Document

To create an interactive grid with form by using the Create Page Wizard:

1. Go to the application home page and click **Create Page.** The Create Page Wizard opens.
2. Select **Report.**
3. Click **Interactive Grid.**
4. On the Create Interactive Grid page, enter *Project Master Document* for Report **Page Name**, select **Breadcrumbs**, and click **Next** (screenshot 2a).

Creating an Interactive Grid (without Form)

3a



Create Interactive Grid

Report Source

Editing Enabled: Yes | No

* Source Type: ● Table ○ SQL Query

* Table / View Owner: PTS

* Table / View Name: PROJECT_DOCUMENTS (table)

Primary Key Column: DOCUMENT_ID (Number)

Primary Key Column 2: - Select Column -

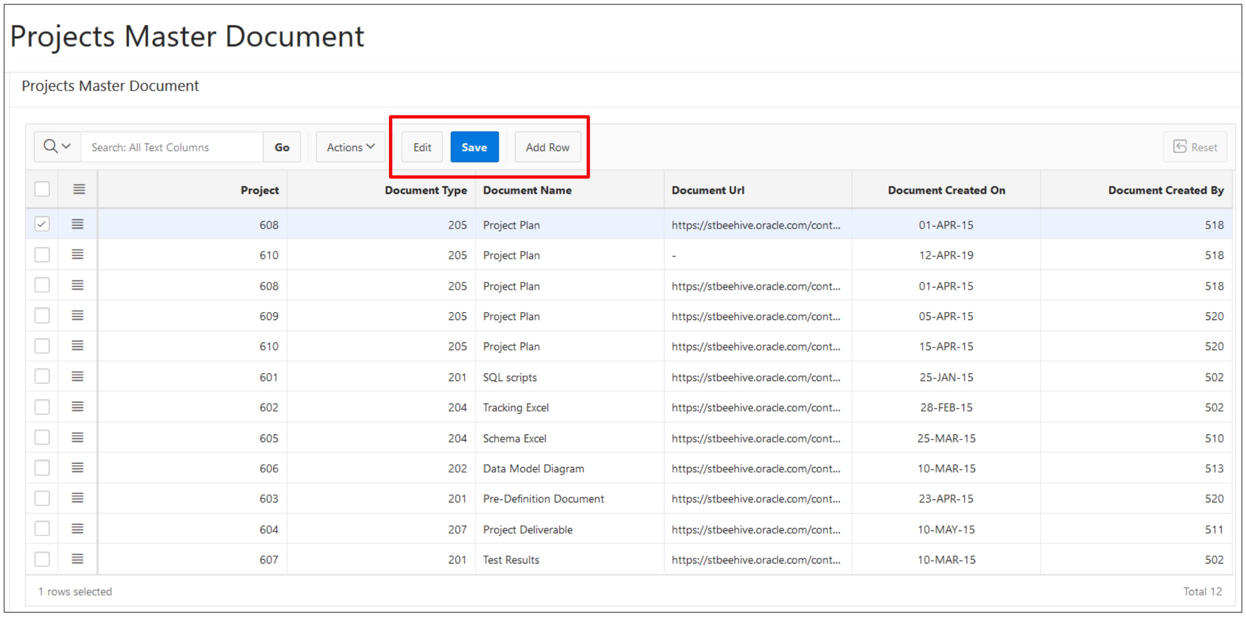
Use User Interface Defaults: ● Yes ○ No

ORACLE

5. Under Navigation menu, click **Select a new navigation menu entry** and click **Next.**
6. In the Report Source section of the wizard, enter the following details (screenshot 3):
   - **Editing Enabled:** Select **Yes**
   - **Source Type:** Click **Table**
   - **Table/View Name:** `PROJECT_DOCUMENTS (table)`
   - **Primary Key Column:** `DOCUMENT_ID (Number)`
   - **Secondary Key Column:** Leave it blank
   - **Use User Interface Defaults: Yes** (default)
   - **Columns:** Click the double right arrow to include all the columns in the report
7. Click Create.
8. In Page Designer, click Save and Run.

The interactive report Project Master Document with the form Document Details is now created.

Interactive Grid (without Form)



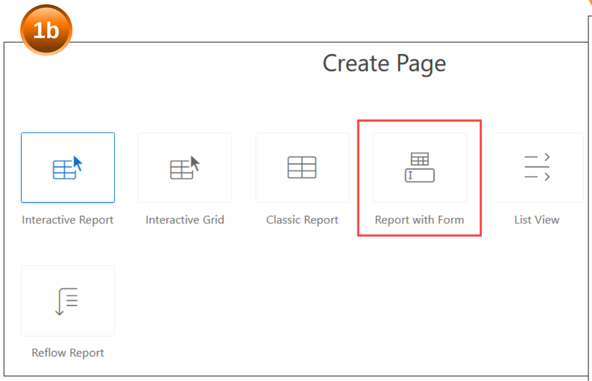
☐	Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
☑	608	205	Project Plan	https://stbeehive.oracle.com/cont...	01-APR-15	518
☐	610	205	Project Plan	-	12-APR-19	518
☐	608	205	Project Plan	https://stbeehive.oracle.com/cont...	01-APR-15	518
☐	609	205	Project Plan	https://stbeehive.oracle.com/cont...	05-APR-15	520
☐	610	205	Project Plan	https://stbeehive.oracle.com/cont...	15-APR-15	520
☐	601	201	SQL scripts	https://stbeehive.oracle.com/cont...	25-JAN-15	502
☐	602	204	Tracking Excel	https://stbeehive.oracle.com/cont...	28-FEB-15	502
☐	605	204	Schema Excel	https://stbeehive.oracle.com/cont...	25-MAR-15	510
☐	606	202	Data Model Diagram	https://stbeehive.oracle.com/cont...	10-MAR-15	513
☐	603	201	Pre-Definition Document	https://stbeehive.oracle.com/cont...	23-APR-15	520
☐	604	207	Project Deliverable	https://stbeehive.oracle.com/cont...	10-MAY-15	511
☐	607	201	Test Results	https://stbeehive.oracle.com/cont...	10-MAR-15	502

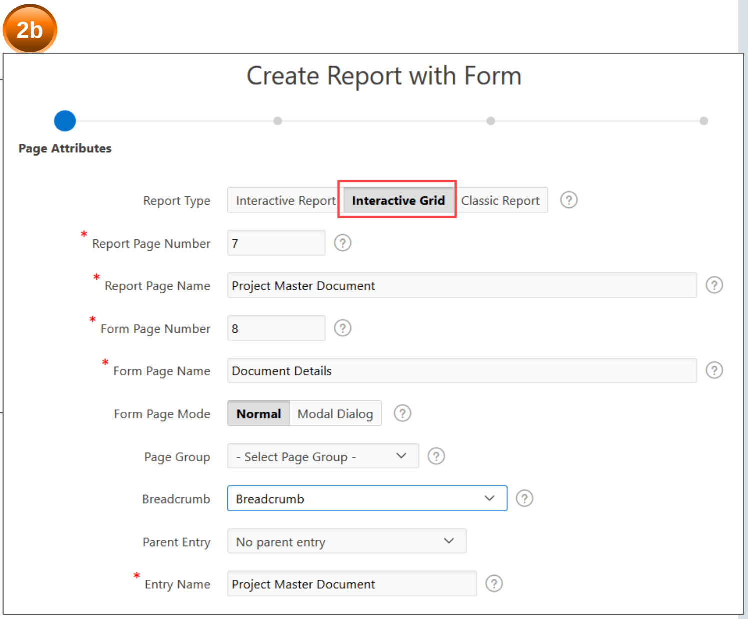
1 rows selected

Total 12

The slide shows the *Project Master Document* interactive grid that you just created. You can add and edit the rows directly on the page. To edit any cell in the grid, place your cursor on the cell and click **Edit.** To add a row, click **Add Row.**

Creating an Interactive Grid (with Form)





Report Type: Interactive Report | **Interactive Grid** | Classic Report

* Report Page Number: 7

* Report Page Name: Project Master Document

* Form Page Number: 8

* Form Page Name: Document Details

Form Page Mode: **Normal** | Modal Dialog

Page Group: - Select Page Group -

Breadcrumb: Breadcrumb

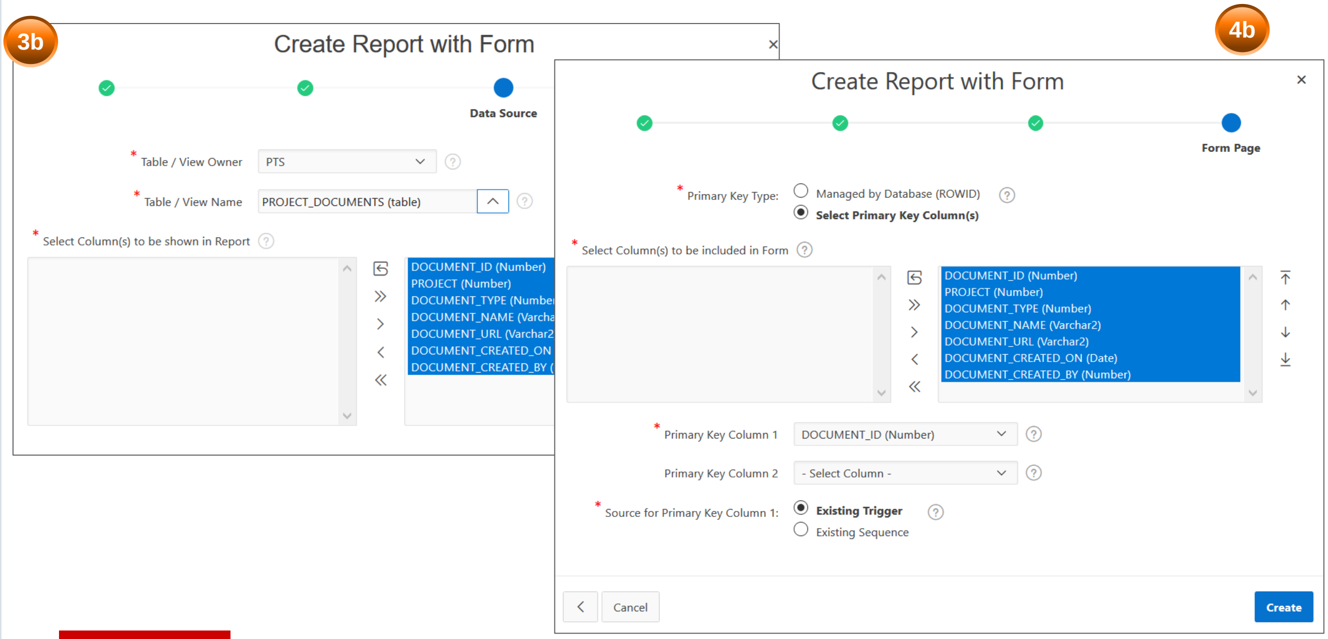
Parent Entry: No parent entry

* Entry Name: Project Master Document

To create an interactive grid with form by using the Create Page Wizard:

1. Go to the application home page and click Create Page. The Create Page Wizard opens.
2. Select Report.
3. Click Report with Form (screenshot 1b)
4. Click Interactive Grid.
5. On the Create Interactive Grid page, enter Project Master Document for Report Page Name and Document Details for Form Page Name and select Breadcrumbs and click Next (screenshot 2b).

Creating an Interactive Grid



Create Report with Form

Data Source

* Table / View Owner: PTS

* Table / View Name: PROJECT_DOCUMENTS (table)

* Select Column(s) to be shown in Report

DOCUMENT_ID (Number)
PROJECT (Number)
DOCUMENT_TYPE (Numbe
DOCUMENT_NAME (Varcha
DOCUMENT_URL (Varchar2
DOCUMENT_CREATED_ON
DOCUMENT_CREATED_BY (

Create Report with Form

Form Page

* Primary Key Type: ○ Managed by Database (ROWID) ● **Select Primary Key Column(s)**

* Select Column(s) to be included in Form

DOCUMENT_ID (Number)
PROJECT (Number)
DOCUMENT_TYPE (Number)
DOCUMENT_NAME (Varchar2)
DOCUMENT_URL (Varchar2)
DOCUMENT_CREATED_ON (Date)
DOCUMENT_CREATED_BY (Number)

* Primary Key Column 1: DOCUMENT_ID (Number)

Primary Key Column 2: - Select Column -

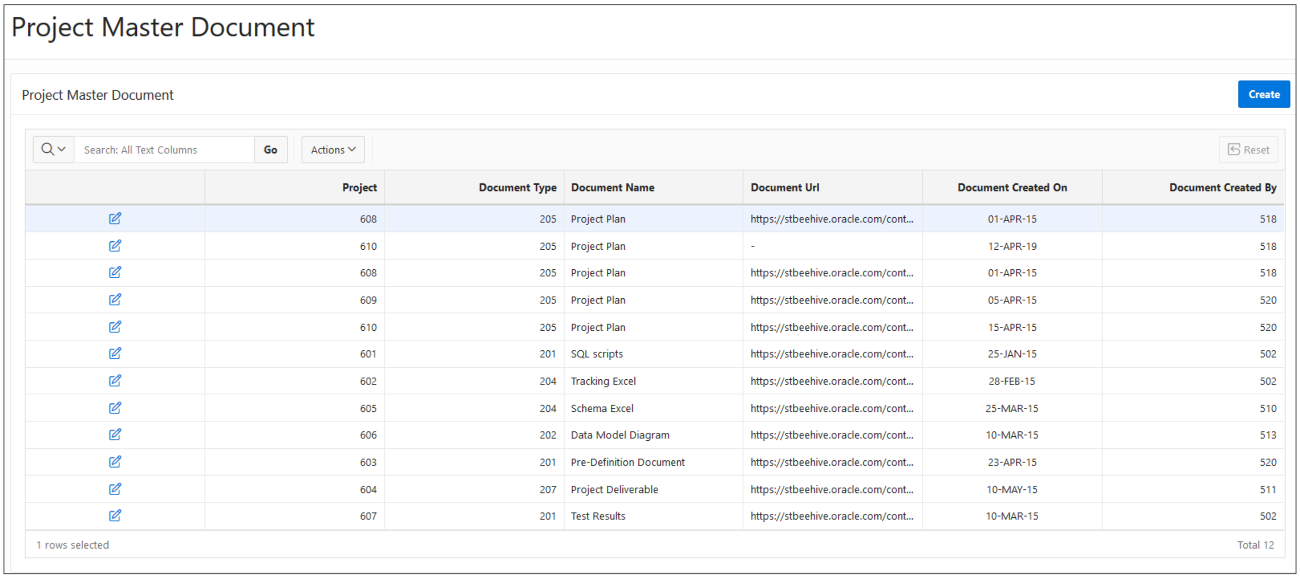
* Source for Primary Key Column 1: ● **Existing Trigger** ○ Existing Sequence

ORACLE

6. Under Navigation menu, click **Select a new navigation menu entry** and **click Next**.
7. In the Data Source section of the wizard, enter the following details (screenshot 3b):
   - **Table/View Owner:** `PTS`
   - **Table/View Name:** `PROJECT_DOCUMENTS (table)`
   - **Columns:** Click the double-right arrow to include all the columns in the report
8. Click Next.
9. In the Form Page section of the wizard, enter the following (screenshot 4b):
   - **Primary Key Type:** Select **Select Primary Key Column(s)**.
   - In the **Select Columns** section, click the double-right arrow to include all the columns in the form.
   - **Primary Key Column 1:** Select `DOCUMENT_ID (Number)`
     Note: The rows in the table must be uniquely identified. You identify the rows by using either a primary key defined on the table or the ROWID pseudo column. In this case, when you select `DOCUMENT_ID` as the primary key column, then the `DOCUMENT_ID` column in the table uniquely identifies the row.
   - **Source for Primary Key Column 1:** Select **Existing Trigger.**
10. Click Create.
11. In Page Designer, click Save and Run.

The interactive report Project Master Document with the form Document Details is now created.

Interactive Grid (with Form)



Project Master Document

Project Master Document

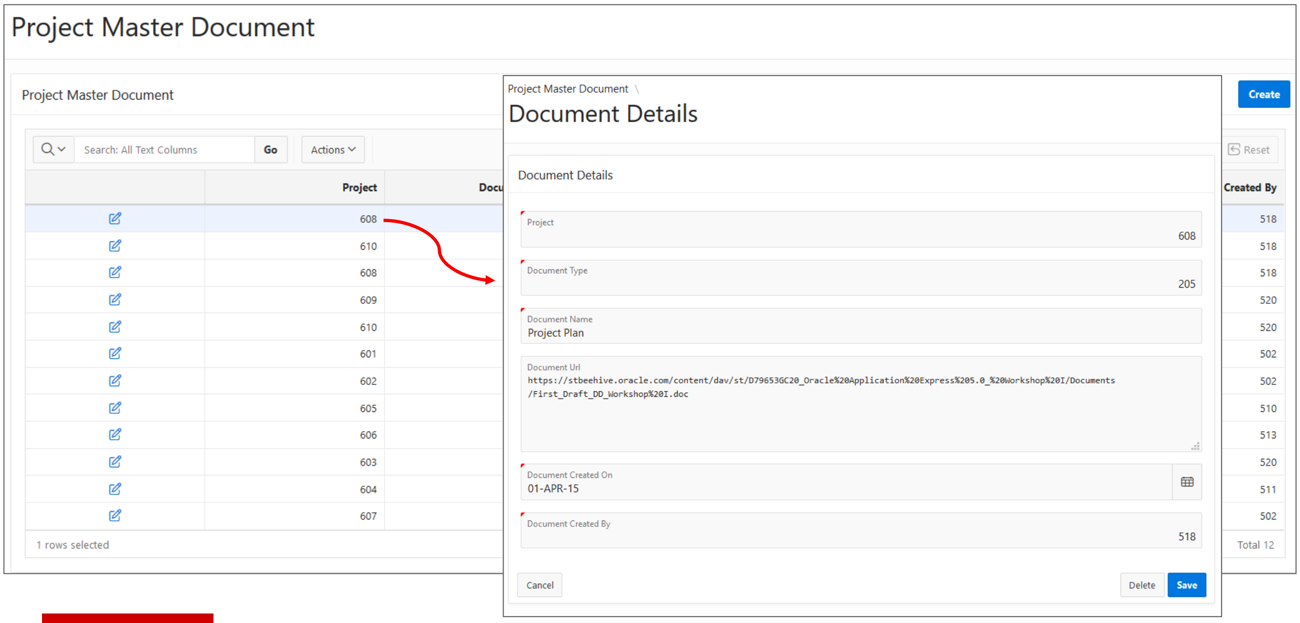
	Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
✎	608	205	Project Plan	https://stbeehive.oracle.com/cont...	01-APR-15	518
✎	610	205	Project Plan	-	12-APR-19	518
✎	608	205	Project Plan	https://stbeehive.oracle.com/cont...	01-APR-15	518
✎	609	205	Project Plan	https://stbeehive.oracle.com/cont...	05-APR-15	520
✎	610	205	Project Plan	https://stbeehive.oracle.com/cont...	15-APR-15	520
✎	601	201	SQL scripts	https://stbeehive.oracle.com/cont...	25-JAN-15	502
✎	602	204	Tracking Excel	https://stbeehive.oracle.com/cont...	28-FEB-15	502
✎	605	204	Schema Excel	https://stbeehive.oracle.com/cont...	25-MAR-15	510
✎	606	202	Data Model Diagram	https://stbeehive.oracle.com/cont...	10-MAR-15	513
✎	603	201	Pre-Definition Document	https://stbeehive.oracle.com/cont...	23-APR-15	520
✎	604	207	Project Deliverable	https://stbeehive.oracle.com/cont...	10-MAY-15	511
✎	607	201	Test Results	https://stbeehive.oracle.com/cont...	10-MAR-15	502

1 rows selected

Total 12

The slide shows the Project Master Document interactive grid that you just created. Note that you cannot edit the rows directly on the page. It does not have the Add, Edit, and Save buttons to edit the grid directly. In slide 22, you will learn how to make this interactive grid editable so that you can edit the rows directly on the grid. You can also

Interactive Grid (with Form)



Project Master Document

Project Master Document

	Project	Docu	Created By
✎	608		518
✎	610		518
✎	608		518
✎	609		520
✎	610		520
✎	601		502
✎	602		502
✎	605		510
✎	606		513
✎	603		520
✎	604		511
✎	607		502

1 rows selected

Total 12

Project Master Document \

Document Details

Document Details

Project: 608

Document Type: 205

Document Name: Project Plan

Document Url: https://stbeehive.oracle.com/content/dav/st/D79653GC20_Oracle%20Application%20Express%205.0_%20Workshop%20I/Documents/First_Draft_DD_Workshop%20I.doc

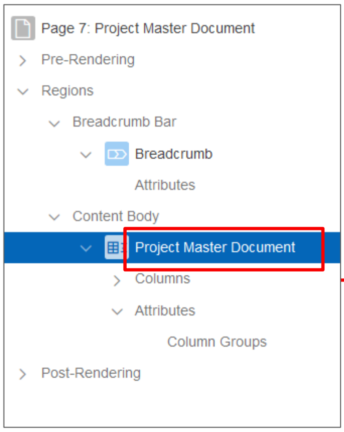
Document Created On: 01-APR-15

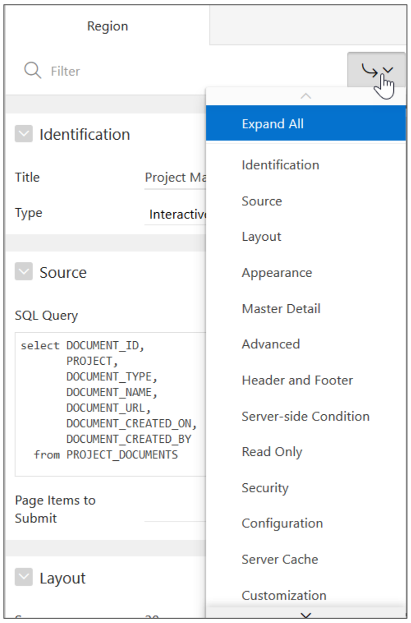
Document Created By: 518

The slide shows the same Project Master Document interactive grid along with the Document Details form that opens when you click the pencil icon. Note that all the fields in the form are editable.

Key Attributes of an Interactive Grid: Region Attribute

Region Attributes define the settings related to the interactive grid region, such as layout and appearance, region conditions, data source for the interactive grid, and so on.





```
select DOCUMENT_ID,
       PROJECT,
       DOCUMENT_TYPE,
       DOCUMENT_NAME,
       DOCUMENT_URL,
       DOCUMENT_CREATED_ON,
       DOCUMENT_CREATED_BY
  from PROJECT_DOCUMENTS
```

ORACLE

You learned how to create an interactive grid. You can modify the behavior and appearance of the interactive grid by editing certain key attributes. For example, you can define and edit the attributes for appearance, layout, data source, download options, button and column labels, user authorization, and so on under three main categories – **Region, Report,** and **Column.** Let us learn more about these key attributes.

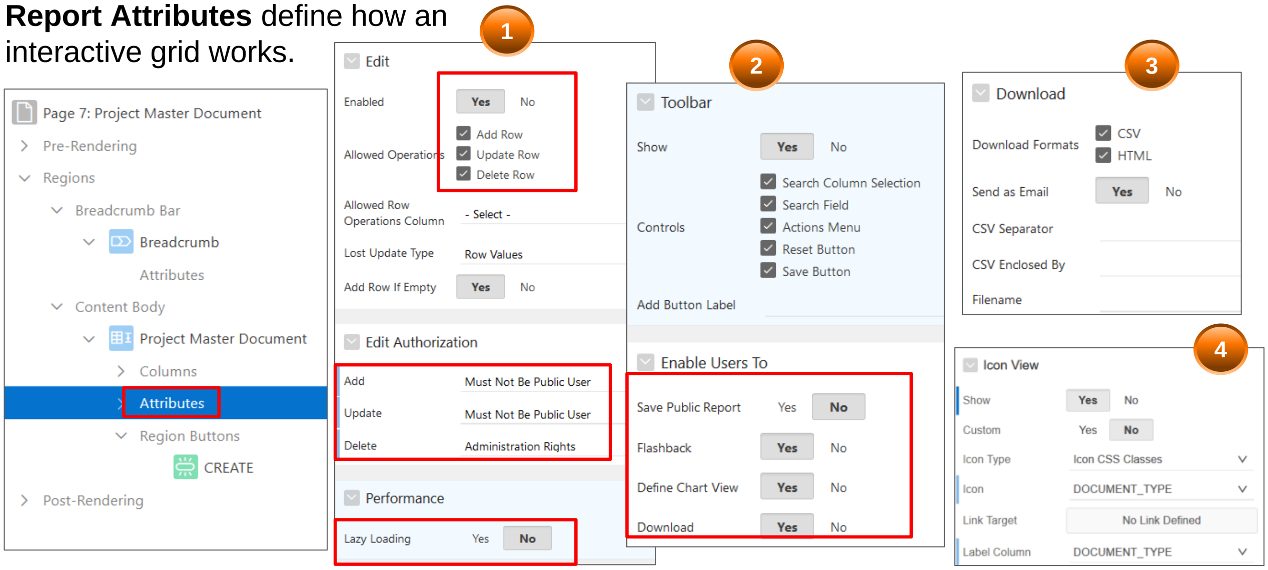
Each application page contains one or more regions. A region is an area on a page that serves as a container for content. You can edit region attributes to alter the SQL source, change the region layout and appearance, define a region display selector, and create region conditions and so on.

In **Region Attributes,** you can:

- Modify the source of the interactive grid
- Define appearance and layout
- Set Header and Footer
- Set Master Detail relation

Key Attributes of an Interactive Grid: Report Attribute

Report Attributes define how an interactive grid works.



Page 7: Project Master Document
- Pre-Rendering
- Regions
  - Breadcrumb Bar
    - Breadcrumb
      - Attributes
  - Content Body
    - Project Master Document
      - Columns
      - Attributes
    - Region Buttons
      - CREATE
- Post-Rendering

Edit

Enabled	[✓ Yes] No
Allowed Operations	☑ Add Row ☑ Update Row ☑ Delete Row
Allowed Row Operations Column	- Select -
Lost Update Type	Row Values
Add Row If Empty	[✓ Yes] No

Edit Authorization

Add	Must Not Be Public User
Update	Must Not Be Public User
Delete	Administration Rights

Performance

Lazy Loading	Yes [✓ No]

Toolbar

Show	[✓ Yes] No
Controls	☑ Search Column Selection ☑ Search Field ☑ Actions Menu ☑ Reset Button ☑ Save Button
Add Button Label	

Enable Users To

Save Public Report	Yes [✓ No]
Flashback	[✓ Yes] No
Define Chart View	[✓ Yes] No
Download	[✓ Yes] No

Download

Download Formats	☑ CSV ☑ HTML
Send as Email	[✓ Yes] No
CSV Separator	
CSV Enclosed By	
Filename	

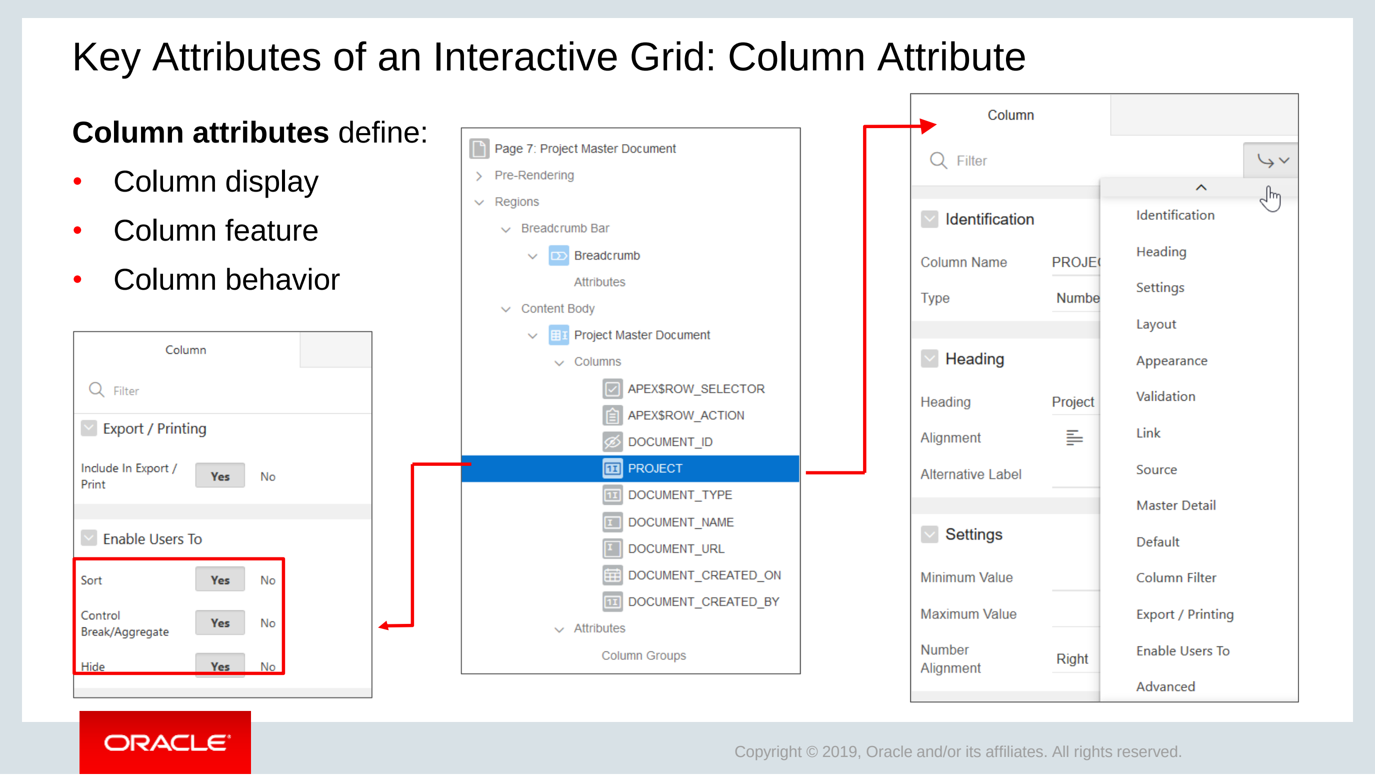
Icon View

Show	[✓ Yes] No
Custom	Yes [✓ No]
Icon Type	Icon CSS Classes
Icon	DOCUMENT_TYPE
Link Target	No Link Defined
Label Column	DOCUMENT_TYPE

Report Attributes control how an interactive grid works. Interactive grid attributes control if end users can edit the underlying data, configure report pagination, create error messages, configure the toolbar and download options, control if users can save public reports, and add Icon and Detail Views.

Under **Report Attributes,** you can:

- Set edit options (screenshot 1)
- Define authorization (screenshot 1)
- Set Lazy Loading performance (screenshot 1)
- Define Toolbar settings (screenshot 2)
- Define permissions for the end user (screenshot 2)
- Define appearance
- Define the download permission and download options (screenshot 2 and 3)
- Define icon view (screenshot 4)



Key Attributes of an Interactive Grid: Column Attribute

Column attributes define:

- Column display
- Column feature
- Column behavior

Column

Export / Printing

Include In Export / Print: Yes No

Enable Users To

Sort: Yes No

Control Break/Aggregate: Yes No

Hide: Yes No

Column

Identification

Column Name: PROJEC

Type: Numbe

Heading

Heading: Project

Alignment

Alternative Label

Settings

Minimum Value

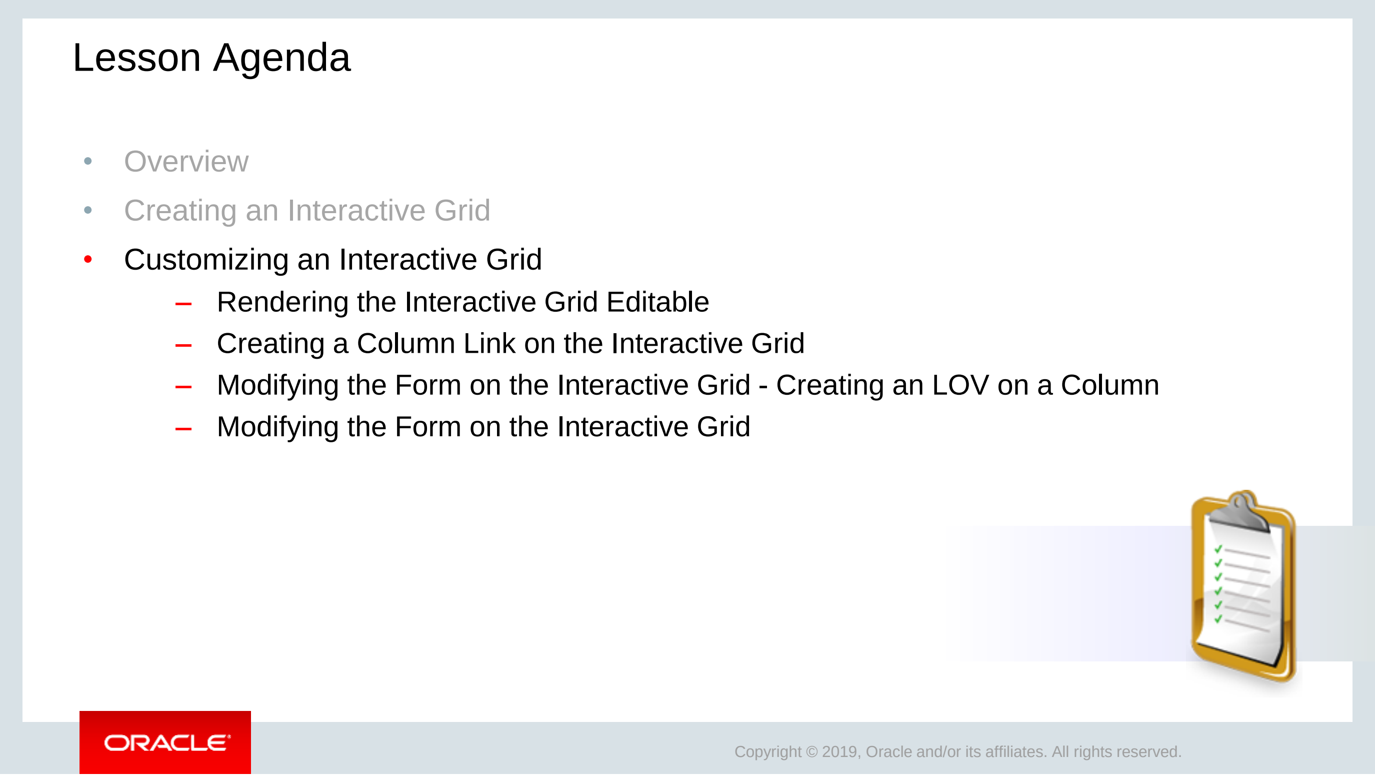
Maximum Value

Number Alignment: Right

Column attributes define the display, features, and behavior of interactive grid columns. You can edit column attributes to alter nearly all aspects of column behavior, including altering the layout and appearance, creating validations, defining column links, creating column filters, and adding support for export and printing.

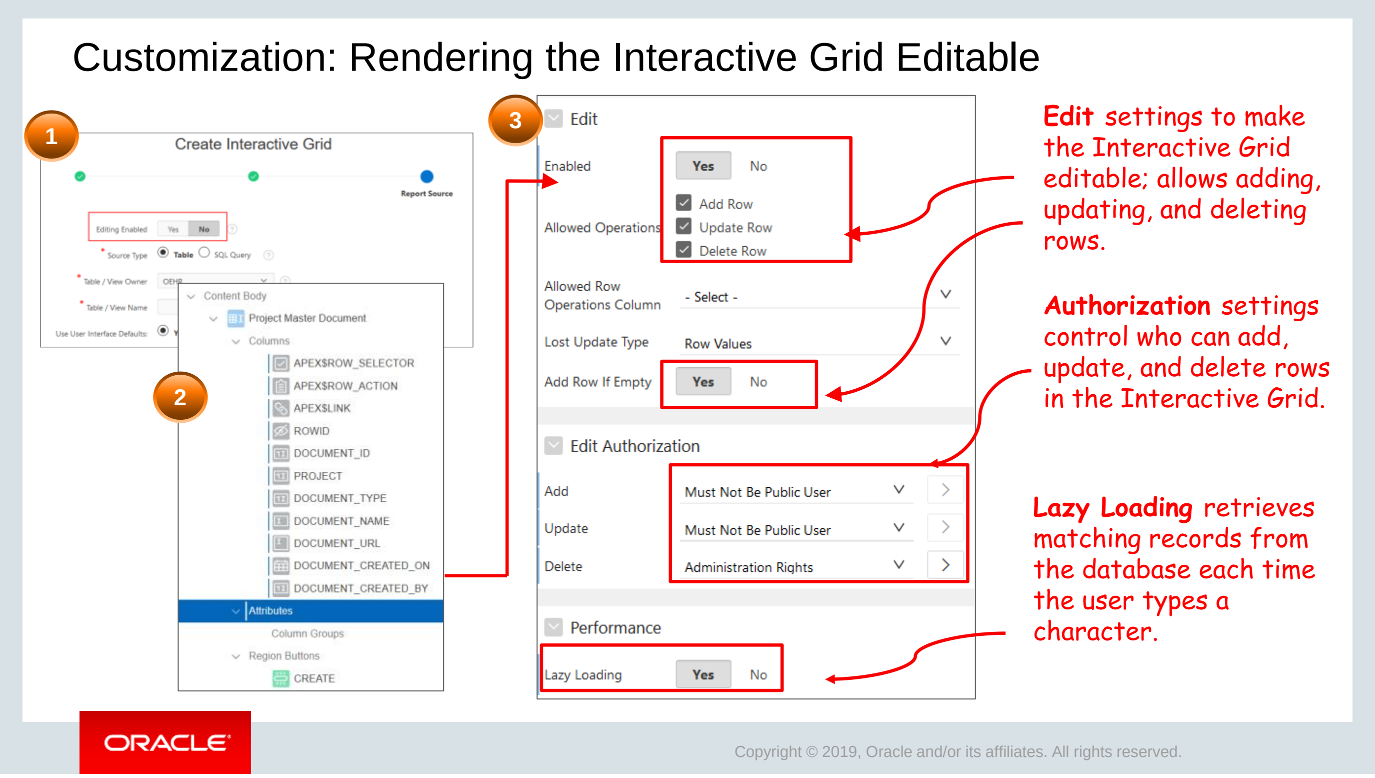
You can define each column of the interactive grid.

- Set Column Filter.
- Create column links.
- Modify column source.
- Set Master Detail settings.
- Set Export/Printing options.
- Set column sort, control, break, and aggregate settings.
- Define validation.
- Edit column headings, layout, and appearance.



Lesson Agenda

- Overview
- Creating an Interactive Grid
- Customizing an Interactive Grid
  - Rendering the Interactive Grid Editable
  - Creating a Column Link on the Interactive Grid
  - Modifying the Form on the Interactive Grid - Creating an LOV on a Column
  - Modifying the Form on the Interactive Grid



Customization: Rendering the Interactive Grid Editable

1 Create Interactive Grid

Report Source

Editing Enabled: Yes [✓ No]

* Source Type: ● Table ○ SQL Query

* Table / View Owner: OEHR

* Table / View Name

Use User Interface Defaults: ●

2

- Content Body
  - Project Master Document
    - Columns
      - APEX$ROW_SELECTOR
      - APEX$ROW_ACTION
      - APEX$LINK
      - ROWID
      - DOCUMENT_ID
      - PROJECT
      - DOCUMENT_TYPE
      - DOCUMENT_NAME
      - DOCUMENT_URL
      - DOCUMENT_CREATED_ON
      - DOCUMENT_CREATED_BY
    - Attributes
    - Column Groups
    - Region Buttons
      - CREATE

3

Edit

Enabled: [✓ Yes] No

Allowed Operations: ☑ Add Row ☑ Update Row ☑ Delete Row

Allowed Row Operations Column: - Select -

Lost Update Type: Row Values

Add Row If Empty: [✓ Yes] No

Edit Authorization

Add: Must Not Be Public User

Update: Must Not Be Public User

Delete: Administration Rights

Performance

Lazy Loading: [✓ Yes] No

Edit settings to make the Interactive Grid editable; allows adding, updating, and deleting rows.

Authorization settings control who can add, update, and delete rows in the Interactive Grid.

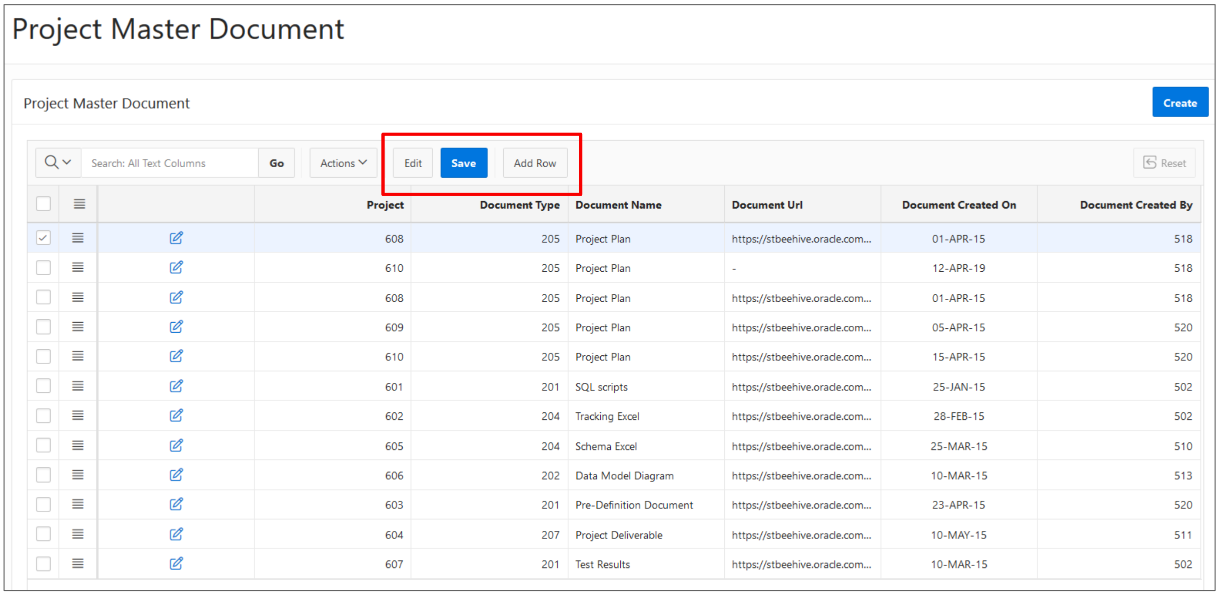
Lazy Loading retrieves matching records from the database each time the user types a character.

When you create an interactive grid, the option **Editing Enabled** is set to **No** (screenshot 1) by default in the wizard. You can modify the attribute settings to make the grid editable after creating the interactive grid. This slide depicts the interactive grid attribute settings in Page Designer that render the grid editable. It shows the **Edit** and **Edit Authorization** settings that define the user who can add, update, and delete rows in the interactive grid.

Open the Project Master Document in Page Designer and make the following changes:

1. Expand the Project Master Document tree on the Rendering tab and click **Attributes.**
2. In the Property Editor, under Edit, set Enabled to Yes.
   a. **Enabled:** Click Yes.
   b. **Allowed Operations:** Select `Add Row, Update Row, Delete Row.`
3. Under Edit Authorization, set the following:
   a. **Add:** Select `Must Not Be Public User.`
   b. **Update:** Select `Must Not Be Public User.`
   c. **Delete:** Select `Administration Rights.` This ensures that only an administrator can delete rows from the interactive grid.
4. Under Performance, select **Yes** for **Lazy Loading.** Lazy Loading retrieves matching records from the database each time the user types a character. Oracle recommends using Lazy Loading when you have a large select list.
5. Click **Save** and **Run.** You will now observe that the interactive grid is now editable.

Interactive Grid: Editable

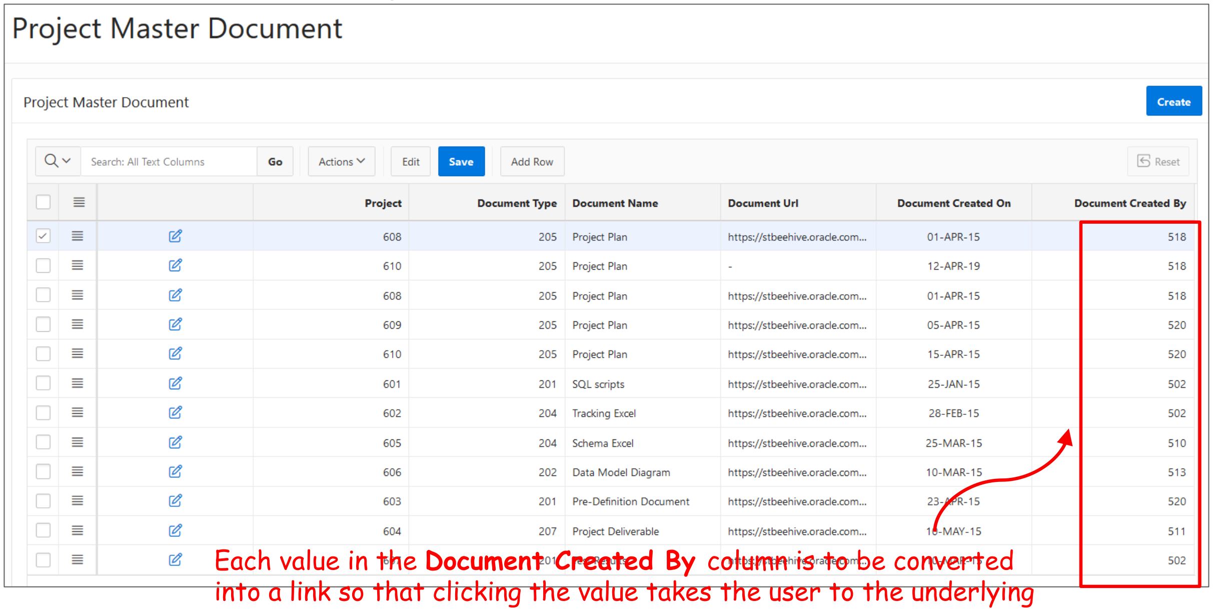


☐	Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
☑	608	205	Project Plan	https://stbeehive.oracle.com...	01-APR-15	518
☐	610	205	Project Plan	-	12-APR-19	518
☐	608	205	Project Plan	https://stbeehive.oracle.com...	01-APR-15	518
☐	609	205	Project Plan	https://stbeehive.oracle.com...	05-APR-15	520
☐	610	205	Project Plan	https://stbeehive.oracle.com...	15-APR-15	520
☐	601	201	SQL scripts	https://stbeehive.oracle.com...	25-JAN-15	502
☐	602	204	Tracking Excel	https://stbeehive.oracle.com...	28-FEB-15	502
☐	605	204	Schema Excel	https://stbeehive.oracle.com...	25-MAR-15	510
☐	606	202	Data Model Diagram	https://stbeehive.oracle.com...	10-MAR-15	513
☐	603	201	Pre-Definition Document	https://stbeehive.oracle.com...	23-APR-15	520
☐	604	207	Project Deliverable	https://stbeehive.oracle.com...	10-MAY-15	511
☐	607	201	Test Results	https://stbeehive.oracle.com...	10-MAR-15	502

ORACLE

The slide shows the *Project Master Document* interactive grid that you just customized to make it editable. Note that the grid now has the **Edit, Save**, and **Add Row** buttons that allow you to edit the grid. It also highlights a cell to show that the cells in the grid can be edited directly on the page.

Customization: Creating Column Links



Project Master Document

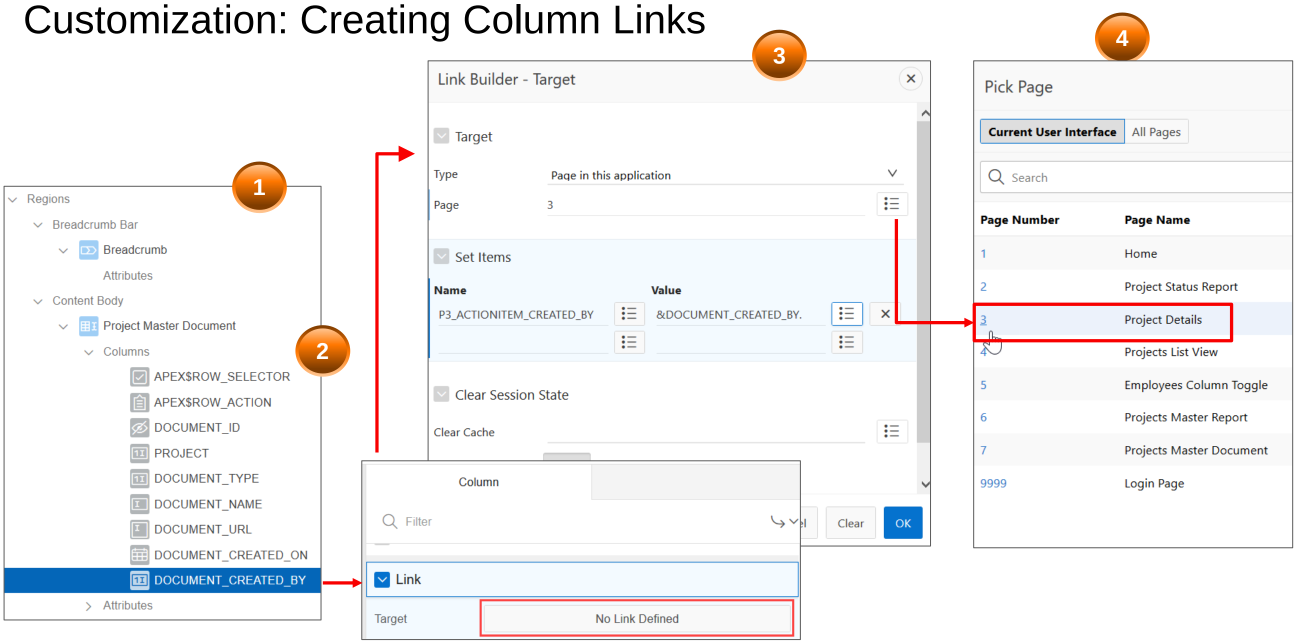
Project Master Document

□			Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
☑	≡	✎	608	205	Project Plan	https://stbeehive.oracle.com...	01-APR-15	518
□	≡	✎	610	205	Project Plan	-	12-APR-19	518
□	≡	✎	608	205	Project Plan	https://stbeehive.oracle.com...	01-APR-15	518
□	≡	✎	609	205	Project Plan	https://stbeehive.oracle.com...	05-APR-15	520
□	≡	✎	610	205	Project Plan	https://stbeehive.oracle.com...	15-APR-15	520
□	≡	✎	601	201	SQL scripts	https://stbeehive.oracle.com...	25-JAN-15	502
□	≡	✎	602	204	Tracking Excel	https://stbeehive.oracle.com...	28-FEB-15	502
□	≡	✎	605	204	Schema Excel	https://stbeehive.oracle.com...	25-MAR-15	510
□	≡	✎	606	202	Data Model Diagram	https://stbeehive.oracle.com...	10-MAR-15	513
□	≡	✎	603	201	Pre-Definition Document	https://stbeehive.oracle.com...	23-APR-15	520
□	≡	✎	604	207	Project Deliverable	https://stbeehive.oracle.com...	10-MAY-15	511
□	≡	✎	607	201	Test Results	https://stbeehive.oracle.com...	10-APR-15	502

Each value in the **Document Created By** column is to be converted into a link so that clicking the value takes the user to the underlying table that contains the relevant details.

In the *Project Master Document,* Steve wants to convert each value in the **Document Created By** column into a link so that clicking the values takes the user to the underlying table that contains the relevant details.

Customization: Creating Column Links



Regions
Breadcrumb Bar
Breadcrumb
Attributes
Content Body
Project Master Document
Columns
APEX$ROW_SELECTOR
APEX$ROW_ACTION
DOCUMENT_ID
PROJECT
DOCUMENT_TYPE
DOCUMENT_NAME
DOCUMENT_URL
DOCUMENT_CREATED_ON
DOCUMENT_CREATED_BY
Attributes

Column

Filter

☑ Link

Target: No Link Defined

Link Builder - Target

☑ Target

Type: Page in this application

Page: 3

☑ Set Items

Name	Value
P3_ACTIONITEM_CREATED_BY	&DOCUMENT_CREATED_BY.

☑ Clear Session State

Clear Cache

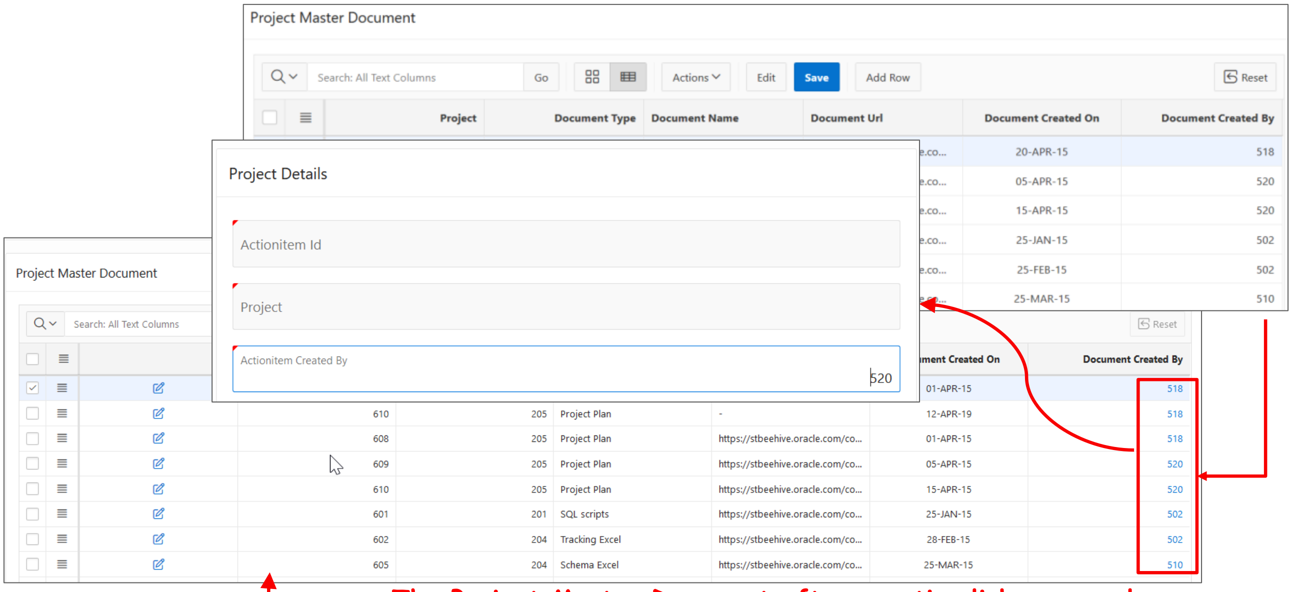
Pick Page

Page Number	Page Name
1	Home
2	Project Status Report
3	Project Details
4	Projects List View
5	Employees Column Toggle
6	Projects Master Report
7	Projects Master Document
9999	Login Page

This slide shows how to link the DOCUMENT CREATED BY column to the PROJECT DETAILS Page in the *Project Master Document* interactive grid. To do this:

1. On the developer toolbar, click **Quick Edit.** The *Project Master Document* opens in Page Designer.
2. On the Rendering tab, expand Columns and click the DOCUMENT_CREATED_BY column.
3. On the right pane, scroll down the column attributes, and under Link, click **No Link Defined.** The Link Builder Target opens.
4. On the Link Builder – Target, enter the following:
   - **Type:** Select **Page in this application.**
   - **Page:** Click the down arrow. In the Pick Page dialog box, click **3 - Project Details.**
   - **Set Items:** For **Name**, select P3_ACTIONITEM_CREATED_BY, and for **Value,** select &DOCUMENT_CREATED_BY
   - Click **OK.**
5. Click **Save** and **Run.**

Customization: Creating Column Links



Project Master Document

		Project	Document Type	Document Name	Document Url	Document Created On	Document Created By
					e.co...	20-APR-15	518
					e.co...	05-APR-15	520
					e.co...	15-APR-15	520
					e.co...	25-JAN-15	502
					e.co...	25-FEB-15	502
					e.co...	25-MAR-15	510

Project Details

Actionitem Id

Project

Actionitem Created By: 520

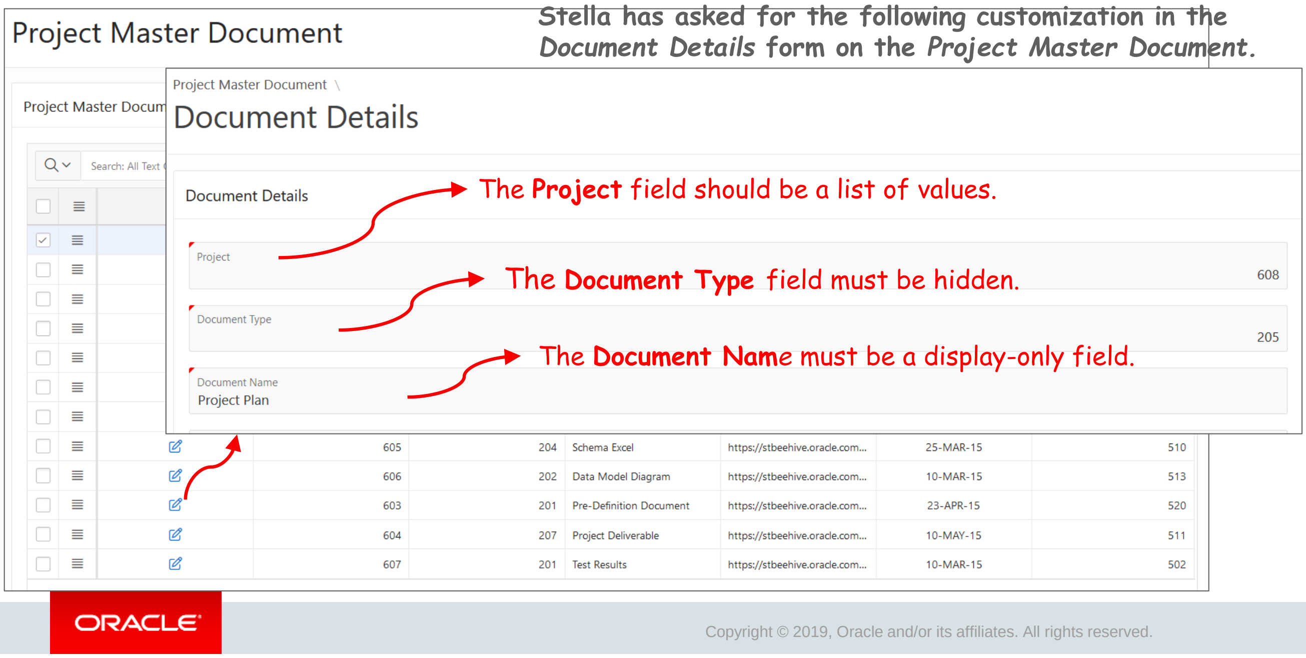
Project Master Document

□							ment Created On	Document Created By
☑	≡	✎					01-APR-15	518
□	≡	✎	610	205	Project Plan	-	12-APR-19	518
□	≡	✎	608	205	Project Plan	https://stbeehive.oracle.com/co...	01-APR-15	518
□	≡	✎	609	205	Project Plan	https://stbeehive.oracle.com/co...	05-APR-15	520
□	≡	✎	610	205	Project Plan	https://stbeehive.oracle.com/co...	15-APR-15	520
□	≡	✎	601	201	SQL scripts	https://stbeehive.oracle.com/co...	25-JAN-15	502
□	≡	✎	602	204	Tracking Excel	https://stbeehive.oracle.com/co...	28-FEB-15	502
□	≡	✎	605	204	Schema Excel	https://stbeehive.oracle.com/co...	25-MAR-15	510

The Project Master Document after creating links on a column

This slide shows the *Project Master Document* interactive grid, both before and after converting the values in the DOCUMENT CREATED BY column to links. The values in the column are linked to the PROJECT DETAILS page (page 6) in the PTS application. In the customized *Project Master Document*, the values in the column are now links. Clicking each link opens the form on PROJECT_ACTIONITEMS.

Customization: Modifying the Form



Stella has asked for the following customization in the *Document Details* form on the *Project Master Document*.

Project Master Document

Project Master Document \

Document Details

Document Details

Project: 608

Document Type: 205

Document Name: Project Plan

The **Project** field should be a list of values.

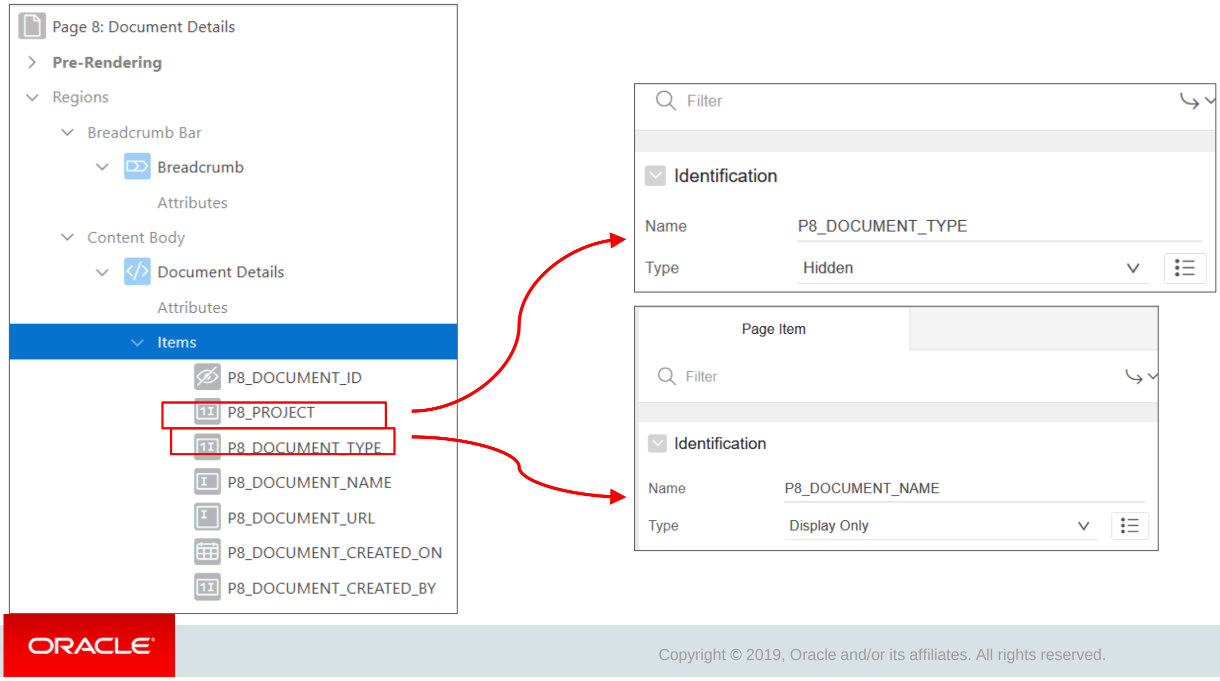
The **Document Type** field must be hidden.

The **Document Name** must be a display-only field.

□	≣	✎	605	204	Schema Excel	https://stbeehive.oracle.com...	25-MAR-15	510
□	≣	✎	606	202	Data Model Diagram	https://stbeehive.oracle.com...	10-MAR-15	513
□	≣	✎	603	201	Pre-Definition Document	https://stbeehive.oracle.com...	23-APR-15	520
□	≣	✎	604	207	Project Deliverable	https://stbeehive.oracle.com...	10-MAY-15	511
□	≣	✎	607	201	Test Results	https://stbeehive.oracle.com...	10-MAR-15	502

The slide shows the *Project Master Document* along with the *Document Details* form on the interactive grid. It shows the customization requirements stated by Stella. Note that all the fields in the form are editable. It is indicated by the red mark in each field.

Customization: Modifying Fields on the Form



Identification

Name P8_DOCUMENT_TYPE

Type Hidden

Identification

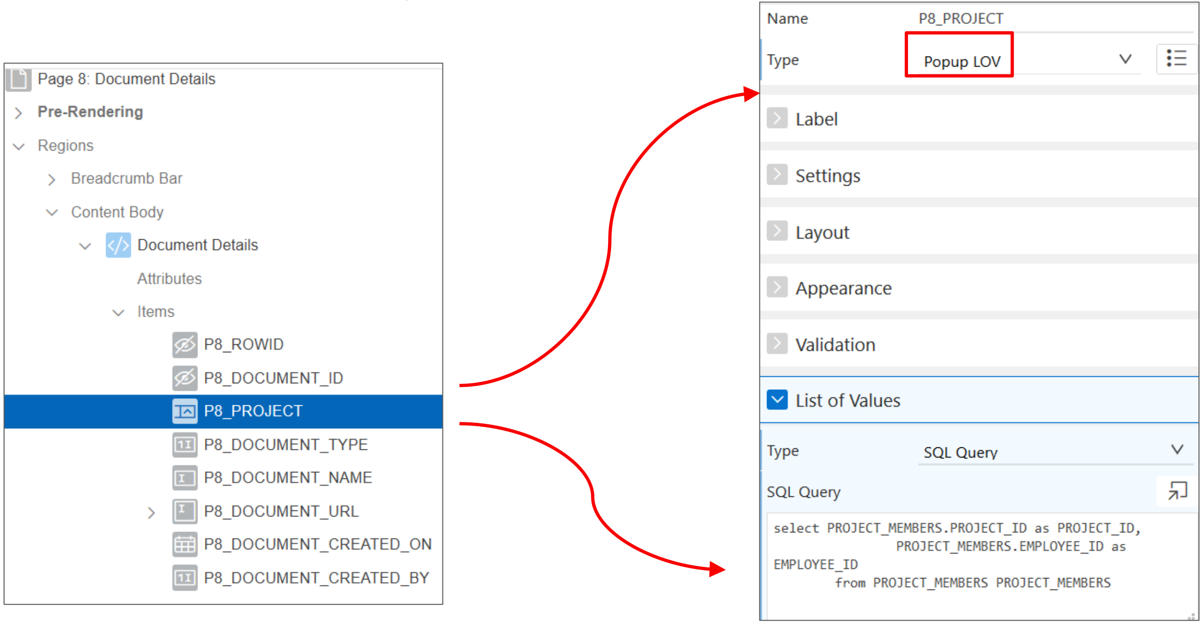
Name P8_DOCUMENT_NAME

Type Display Only

This slide shows the *Document Details* page as opened in Page Designer. It shows the specific attributes for the `P8_DOCUMENT_TYPE` and `P8_DOCUMENT_NAME` columns that you will learn to customize.

1. On the Document Details developer toolbar, click **Quick Edit.** The page opens in Page Designer.
2. To hide the Document Type field:
   a) On the Rendering tab, click `P8_DOCUMENT_TYPE`.
   b) On the column attributes pane on the right, expand **Identification**, and under Type, select **Hidden.**
   c) Click **Save.**
3. To convert the Document Name field to display only:
   a) On the Rendering tab, click `P8_DOCUMENT_NAME`
   b) On the column attributes pane on the right, expand **Identification**, and under Type, select **Display Only.**
   c) Click **Save.**

Customization: Creating a List of Values on a Column



Name: P8_PROJECT

Type: Popup LOV

List of Values

Type: SQL Query

SQL Query

```
select PROJECT_MEMBERS.PROJECT_ID as PROJECT_ID,
                PROJECT_MEMBERS.EMPLOYEE_ID as
EMPLOYEE_ID
        from PROJECT_MEMBERS PROJECT_MEMBERS
```

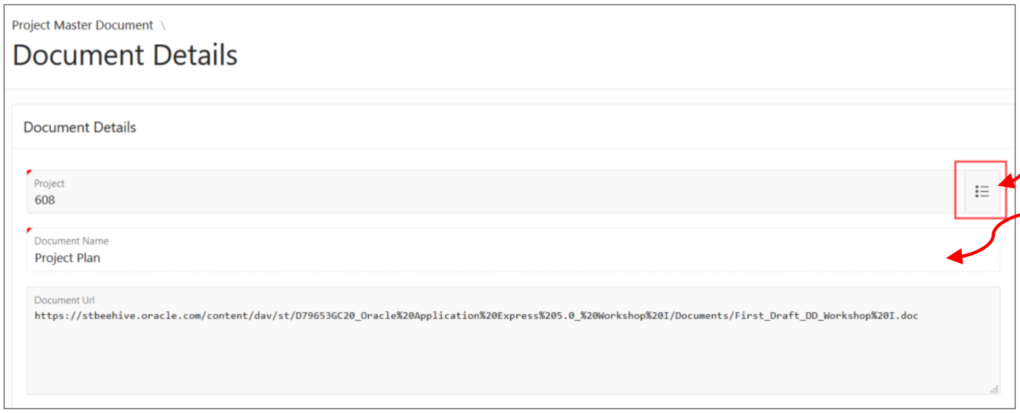
This slide shows the *Document Details* page as opened in Page Designer. It shows the specific attributes for the P8_PROJECT column that you will learn to change from a number field to a list of values.

1. On the **Document Details** developer toolbar, click **Quick Edit.** The page opens in Page Designer.
2. On the **Rendering** tab, expand Items under Document Details and click **P8_Project.**
3. In the **Attributes** pane, under Identification, select Popup LOV for Type.
4. In the **List of Values** section, enter the following:
   - **a) Type:** Select **SQL Query.**
   - **b) SQL Query:** Enter the following:

```
            select PROJECT_MEMBERS.PROJECT_ID as PROJECT_ID,
           PROJECT_MEMBERS.EMPLOYEE_ID as EMPLOYEE_ID
   from PROJECT_MEMBERS PROJECT_MEMBERS
```

5. Click **Save** and **Run.**

Customization: Creating a List of Values on a Column



Project Master Document \

Document Details

Document Details

Project
608

Document Name
Project Plan

Document Url
https://stbeehive.oracle.com/content/dav/st/D79653GC20_Oracle%20Application%20Express%205.0_%20Workshop%20I/Documents/First_Draft_DD_Workshop%20I.doc

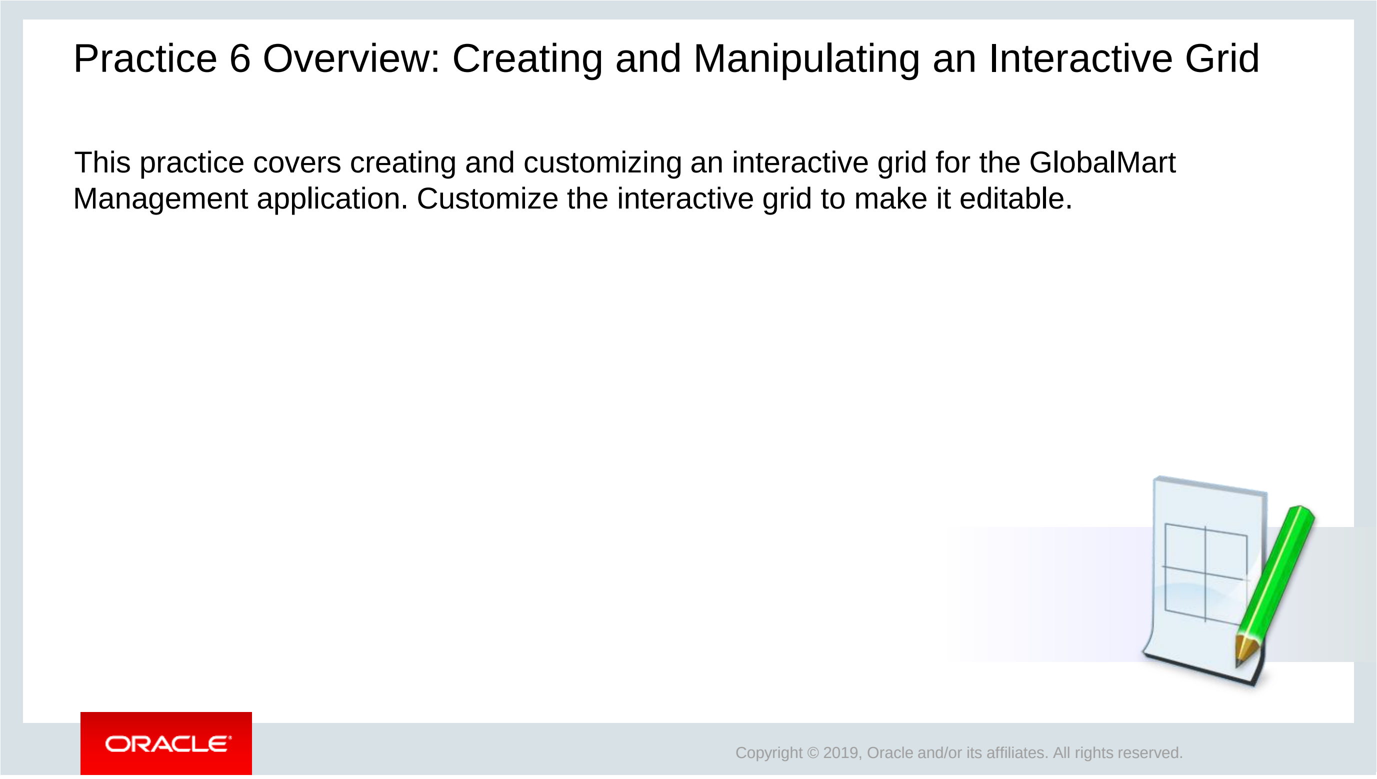
The **Project** field is now a popup list of values.

The **Document Name** field is now a display-only field. It is noneditable.

The **Document ID** field is no longer visible on the form.

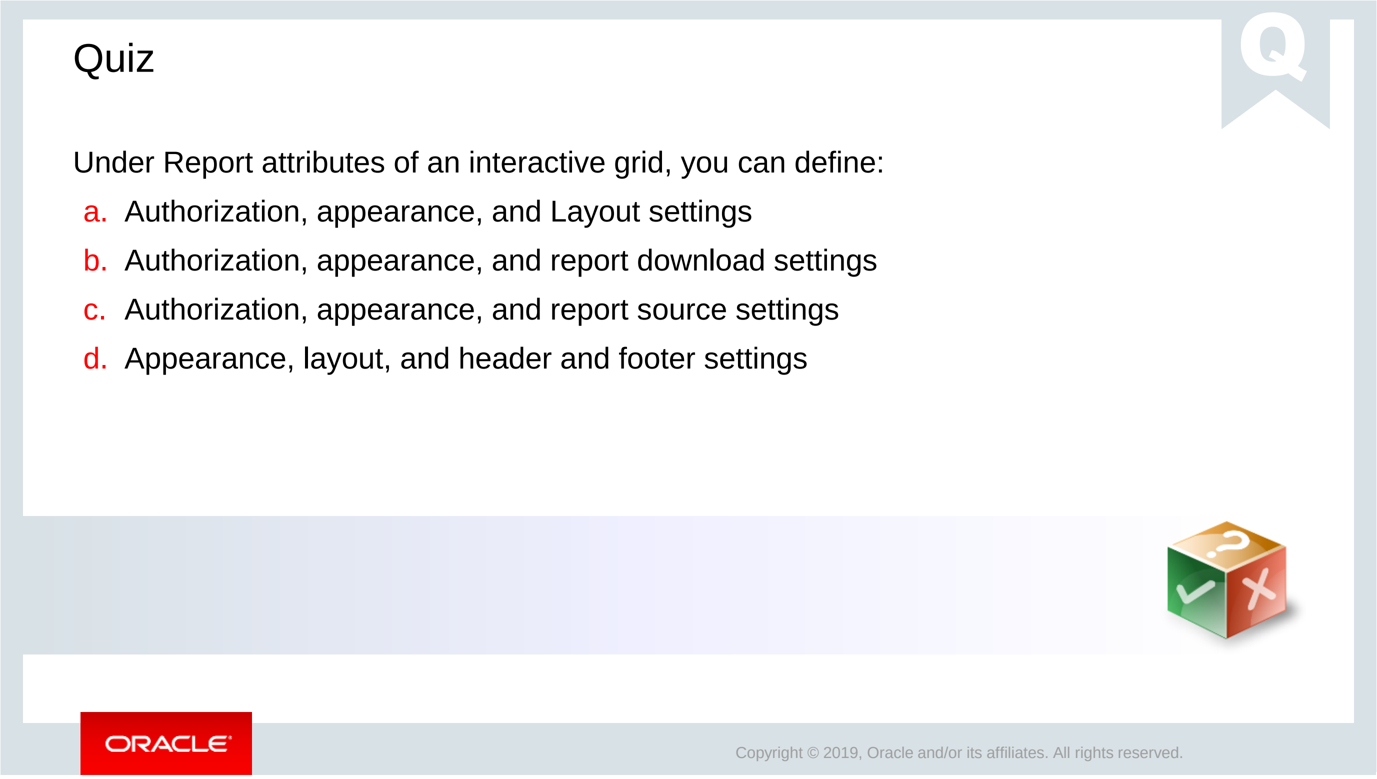
ORACLE

The screenshot in the slide shows the *Document Details* form that you just edited. In this form, the **Project** field is not a pop-up list, the **Document Name** field is not a display-only field, and the **Document ID** field is no longer visible.



Practice 6 Overview: Creating and Manipulating an Interactive Grid

This practice covers creating and customizing an interactive grid for the GlobalMart Management application. Customize the interactive grid to make it editable.



Quiz

Under Report attributes of an interactive grid, you can define:

a. Authorization, appearance, and Layout settings

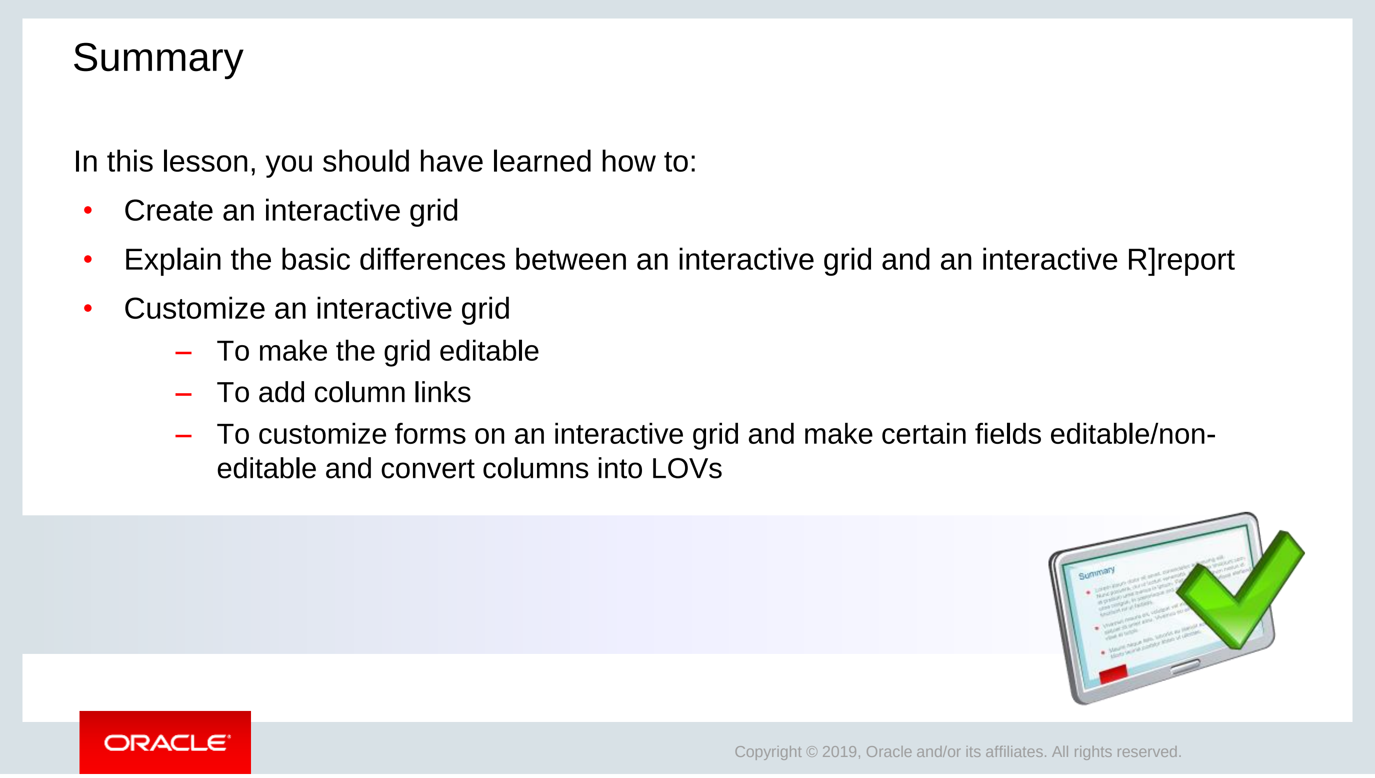
b. Authorization, appearance, and report download settings

c. Authorization, appearance, and report source settings

d. Appearance, layout, and header and footer settings

ORACLE

Answer: b



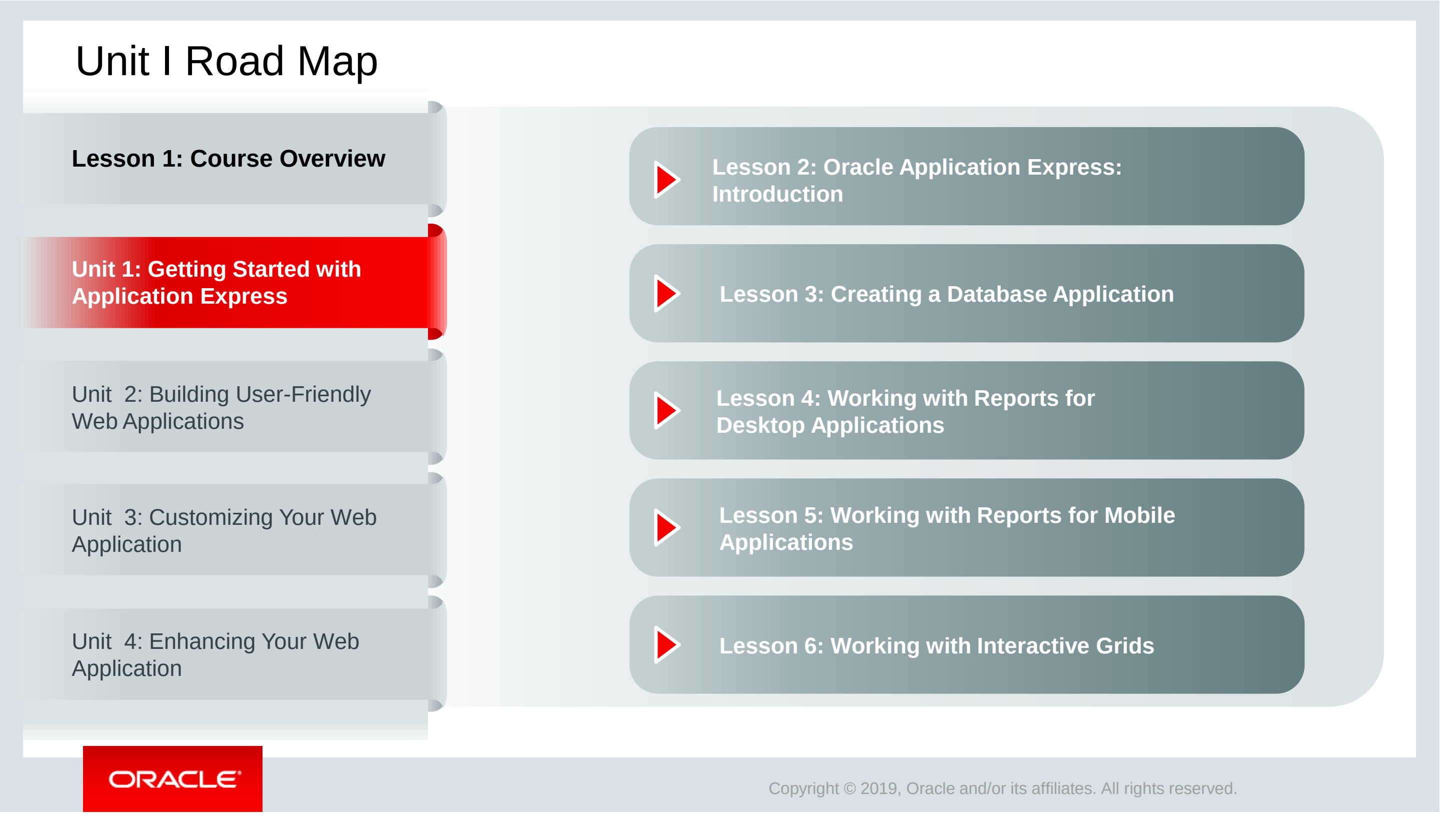
Summary

In this lesson, you should have learned how to:

- Create an interactive grid
- Explain the basic differences between an interactive grid and an interactive R]report
- Customize an interactive grid
  - To make the grid editable
  - To add column links
  - To customize forms on an interactive grid and make certain fields editable/non-editable and convert columns into LOVs

In this lesson, you learned how to create and customize interactive grids.

Unit I Summary: Getting Started with Application Express



Unit I Road Map

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

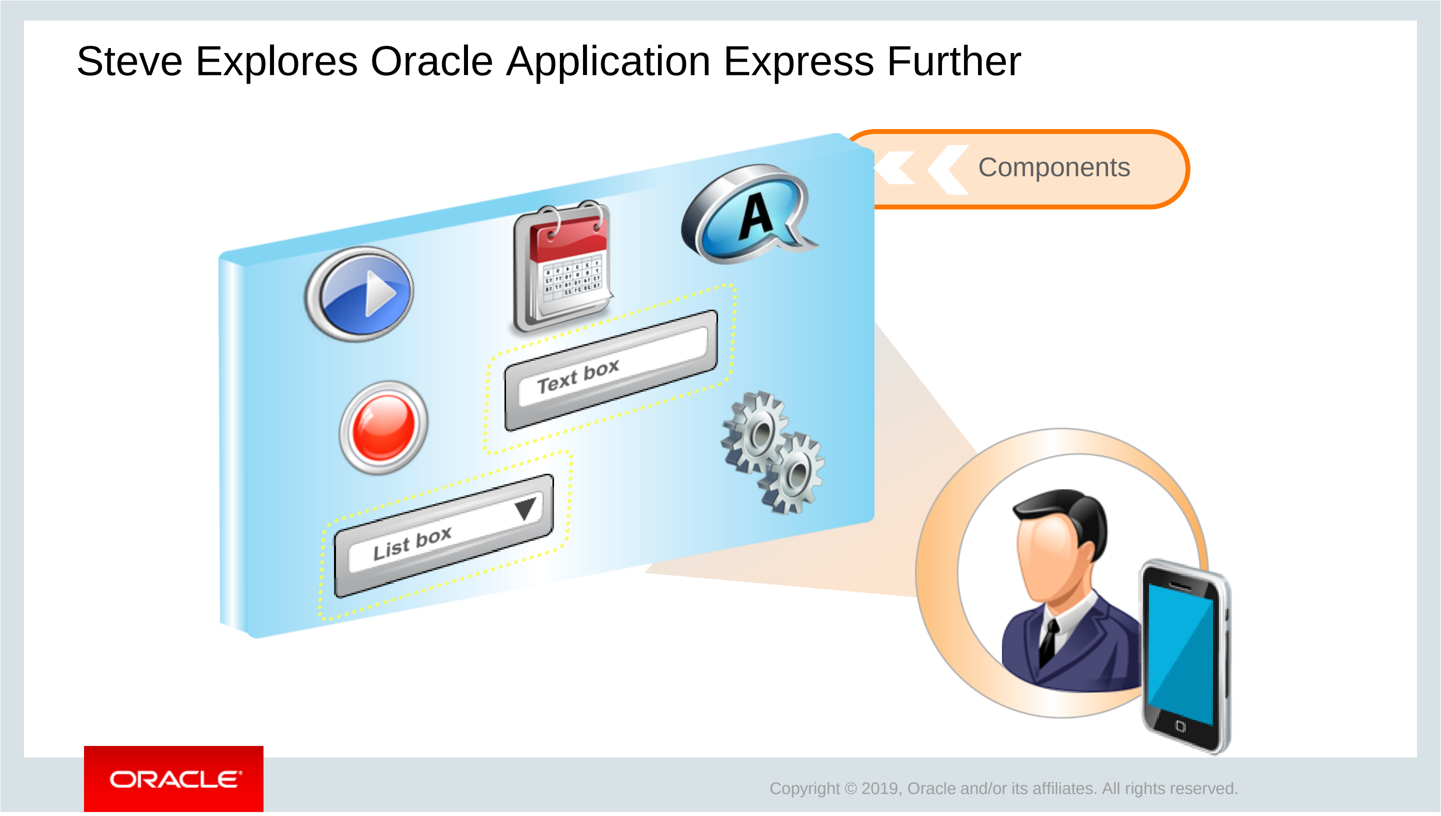
Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- **Lesson 2: Oracle Application Express: Introduction**
- **Lesson 3: Creating a Database Application**
- **Lesson 4: Working with Reports for Desktop Applications**
- **Lesson 5: Working with Reports for Mobile Applications**
- **Lesson 6: Working with Interactive Grids**

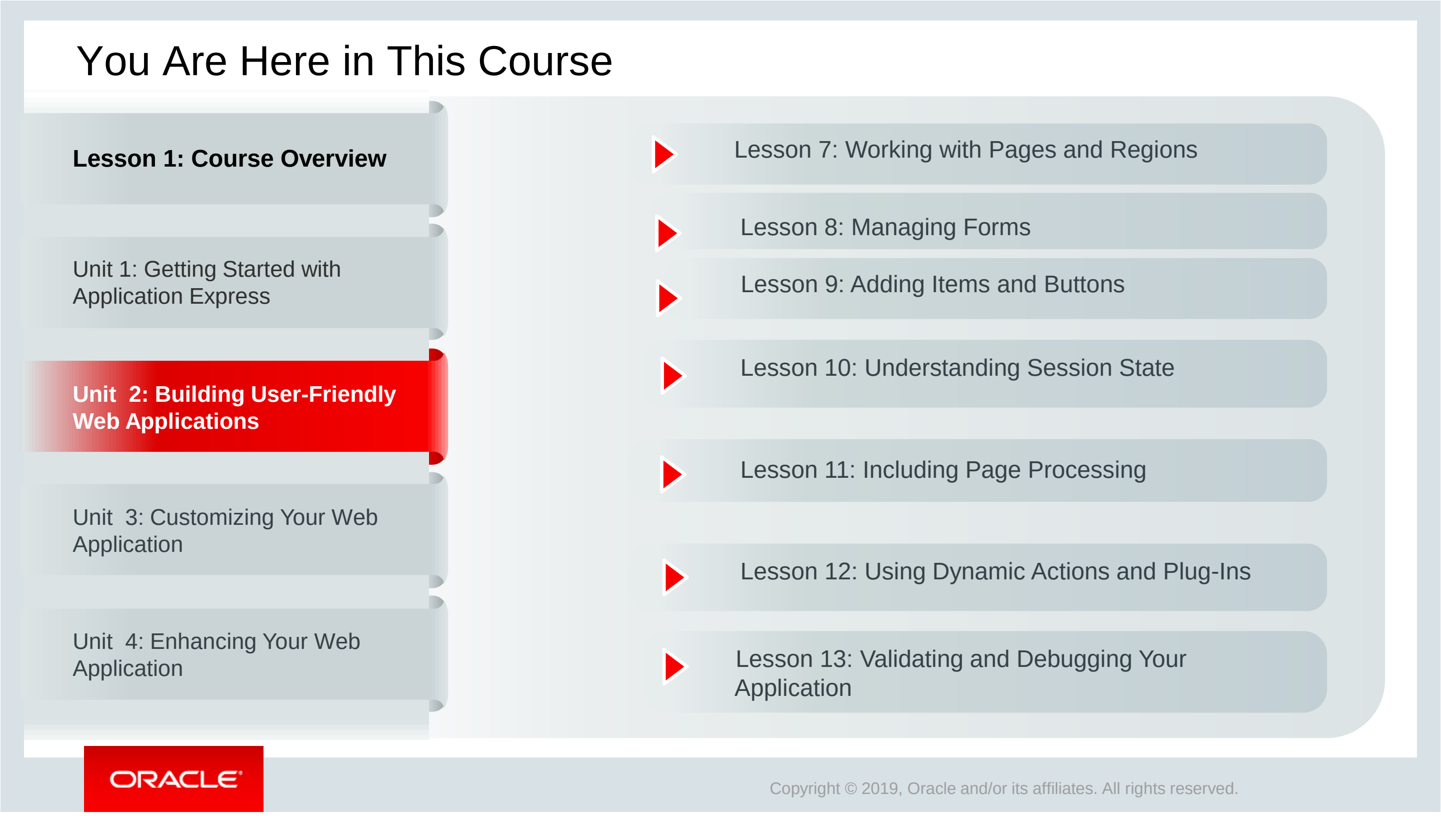
In Unit 1, you completed five topics.

Unit II Introduction: Building User-Friendly Web Applications



Steve Explores Oracle Application Express Further

Steve got some good experience working with Oracle Application Express for developing database applications and has created a *Project Tracking System* (*PTS*) application, which generates required reports. He now wants to add additional user interface screens to *PTS* and make it more user-friendly. To do this, he plans to look deeper into Oracle Application Express Pages, Regions, and other Page Components.



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

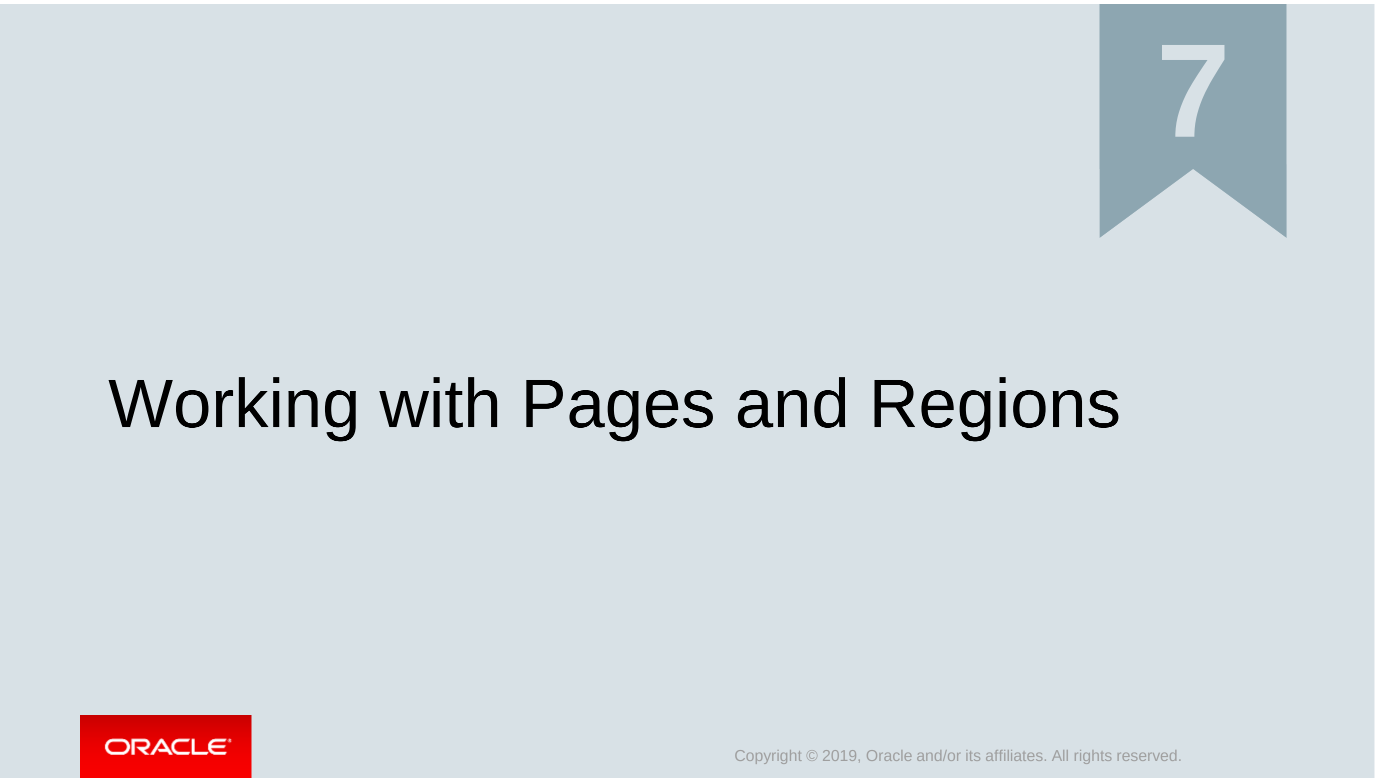
Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

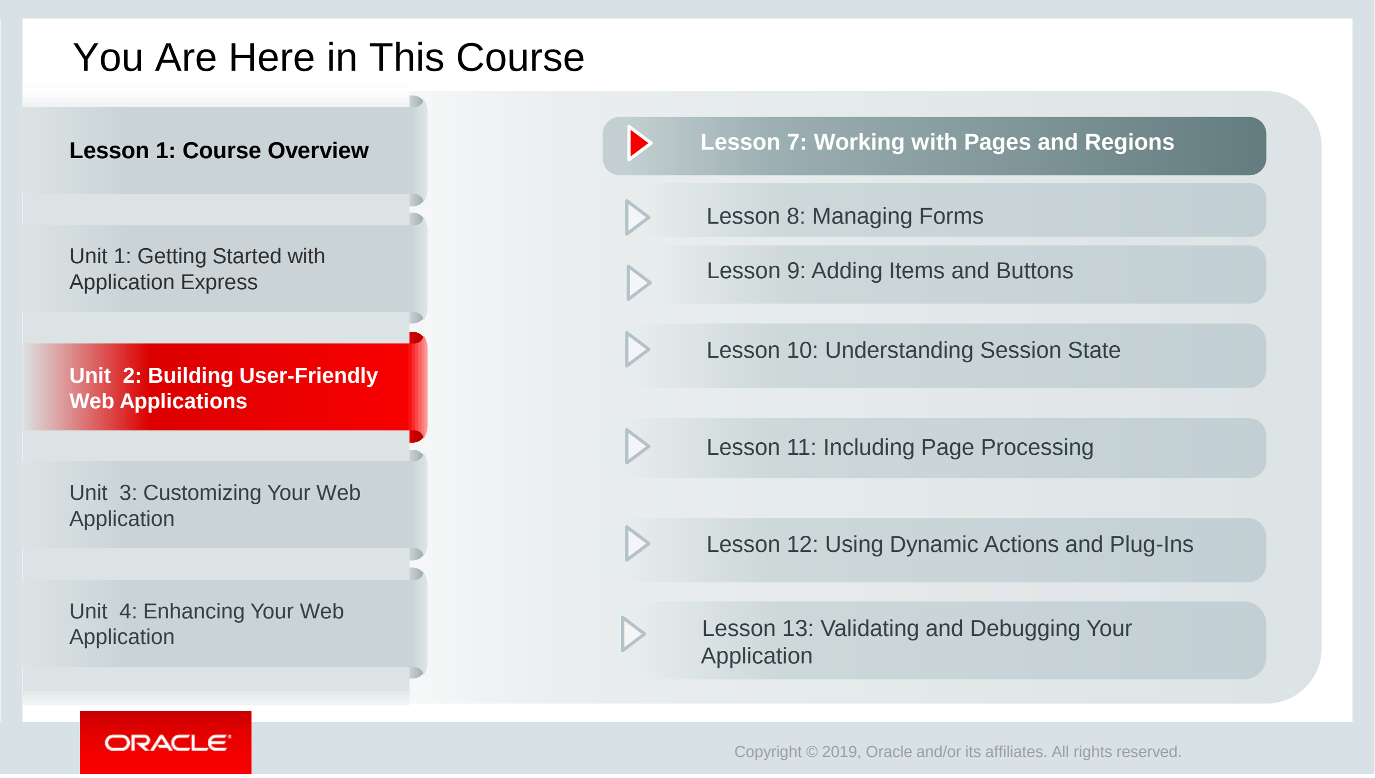
- Lesson 7: Working with Pages and Regions
- Lesson 8: Managing Forms
- Lesson 9: Adding Items and Buttons
- Lesson 10: Understanding Session State
- Lesson 11: Including Page Processing
- Lesson 12: Using Dynamic Actions and Plug-Ins
- Lesson 13: Validating and Debugging Your Application

In Unit 2, you learn to build a user-friendly interface for your application with the help of forms, pages, and regions. You also learn how to add items, buttons, and processing to your pages to support validations and computations and learn about dynamic actions and how to validate and debug your application. This unit is organized into seven lessons. Each lesson has an associated activity guide that allows students to put their learnings into practice.



7

Working with Pages and Regions



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

Lesson 7: Working with Pages and Regions

Lesson 8: Managing Forms

Lesson 9: Adding Items and Buttons

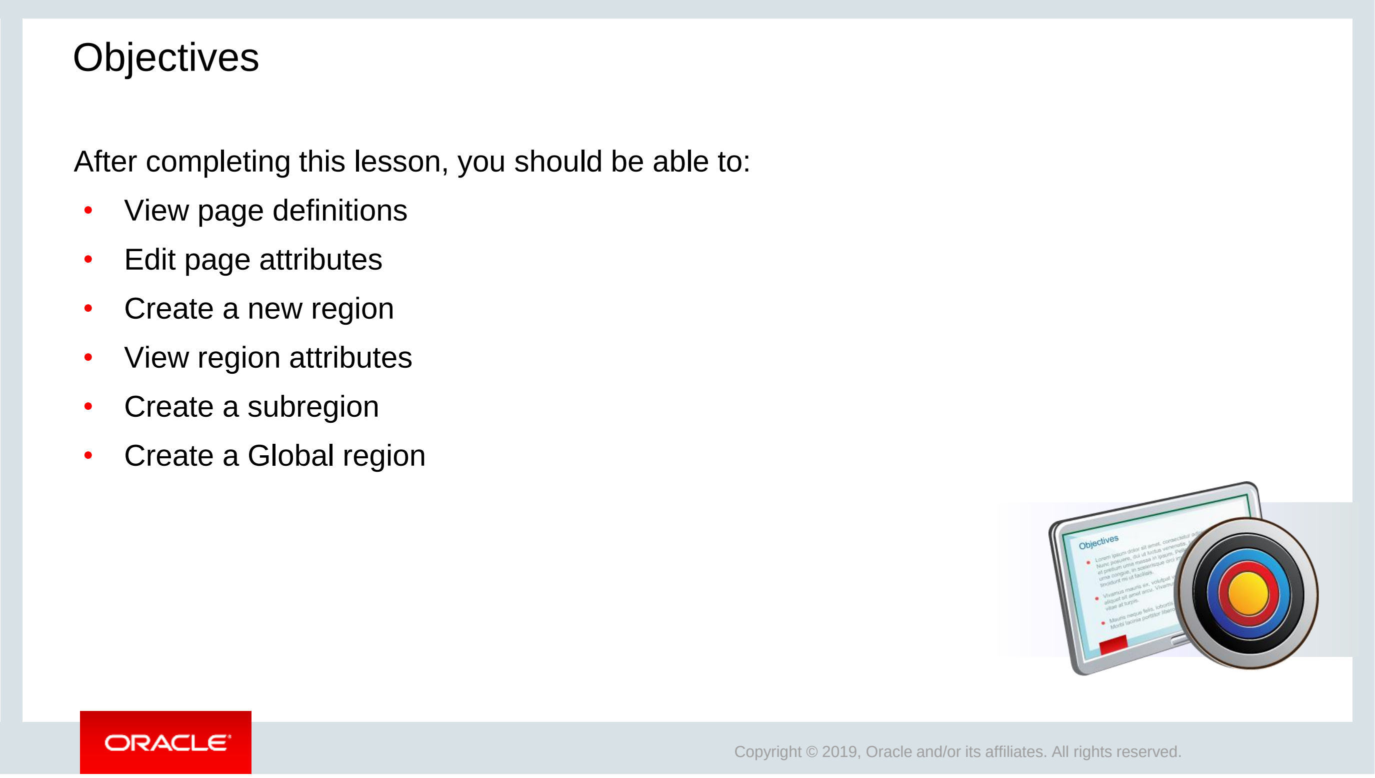
Lesson 10: Understanding Session State

Lesson 11: Including Page Processing

Lesson 12: Using Dynamic Actions and Plug-Ins

Lesson 13: Validating and Debugging Your Application

This slide is a graphical depiction of the course, particularly highlighting Unit 2 - Lesson 7, which is dealt with in these slides.



Objectives

After completing this lesson, you should be able to:

- View page definitions
- Edit page attributes
- Create a new region
- View region attributes
- Create a subregion
- Create a Global region

This lesson shows you how to create pages and regions and how to edit their attributes.

Steve Works with Pages and Regions

Steve thinks of improving the user interface (UI) for his *Project Tracking System (PTS)* application. While going through the features of Oracle Application Express, he discovers that he can create blank pages in the application that can be customized based on his requirements. However, because the *PTS* application is not quite functional, he does not want to try anything on *PTS* right now. Instead, he wants to go through the *Sample Database Application* (which was installed in the lesson titled “Oracle Application Express: Introduction” in this course), because it is a fully functional and editable application.

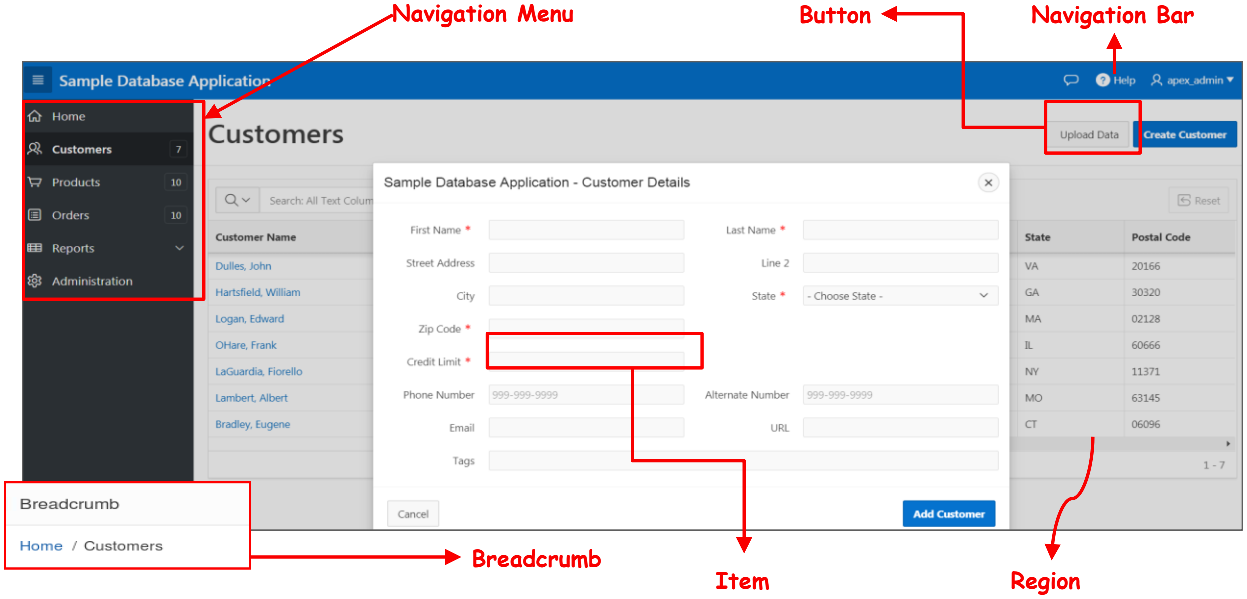
However, he switches to the *PTS* application briefly while working on the Global pages later in this lesson.

Let us see how Steve takes us through the varied options that Oracle Application Express provides via these two applications.

Lesson Agenda

- Introducing Page Definition
  - What Is a Page?
  - Accessing Page Definition
  - Page Modes: Normal, Modal, and Nonmodal
  - What is Page Designer?
  - Common Page Designer UI Elements
  - Editing Page Attributes
  - Running a Page
  - Runtime Developer Toolbar
- Working with Page Regions
- Working with Pages

What Is a Page?



Customer Name	State	Postal Code
Dulles, John	VA	20166
Hartsfield, William	GA	30320
Logan, Edward	MA	02128
OHare, Frank	IL	60666
LaGuardia, Fiorello	NY	11371
Lambert, Albert	MO	63145
Bradley, Eugene	CT	06096

In the lesson titled “Creating a Database Application,” you learned that a page is the basic building block of any application. This slide presents a recap of the different components of a page for you. Now, let’s explore a bit more and understand what a page is in detail.

A page basically contains both user interface elements and application logic and is divided into regions. Each region is a section of a page that contains content. For example, a region can contain a report based on a SQL query, or it can contain static HTML. A region can also contain:

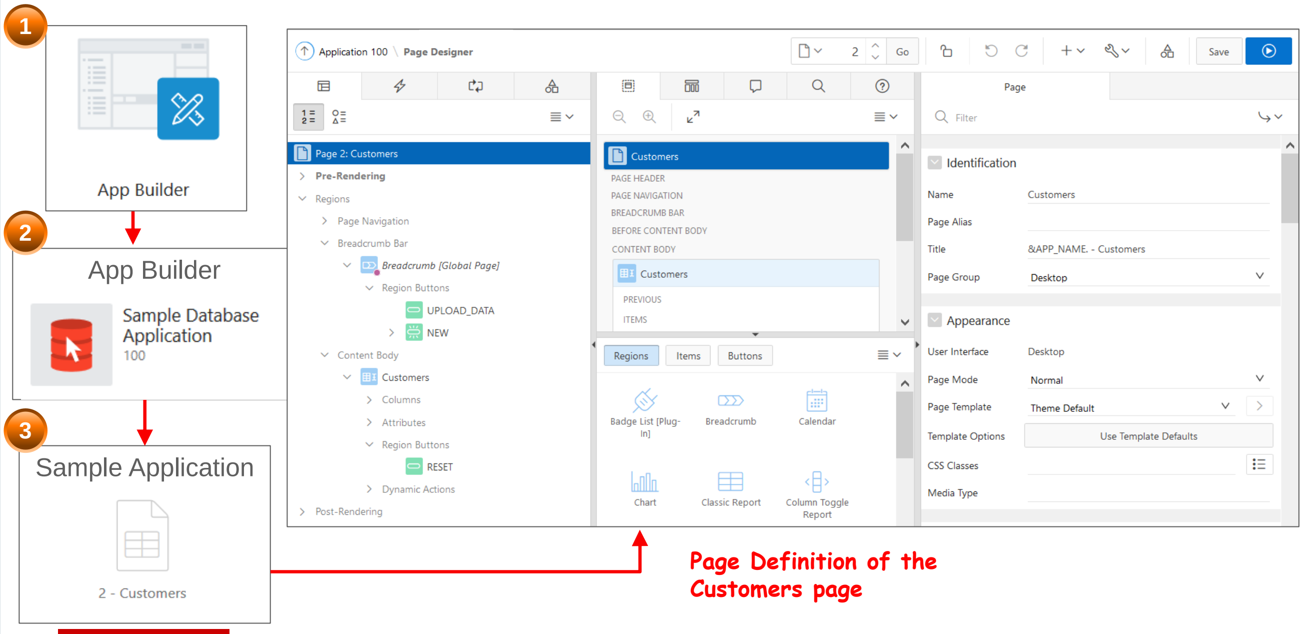
- Items, such as a text field, text area, select list, and check box
- Buttons to direct users to a specific page or URL and also to post and process information
- Breadcrumbs, navigation menu, and a navigation bar to enable navigation

You will learn how to use all these in a later chapter of this unit.

You can easily identify a page in your application by its unique page ID and name and also find all relevant information about a page and its components in a *page definition.*

Let’s see what a *page definition* is in the following slide.

Accessing Page Definition



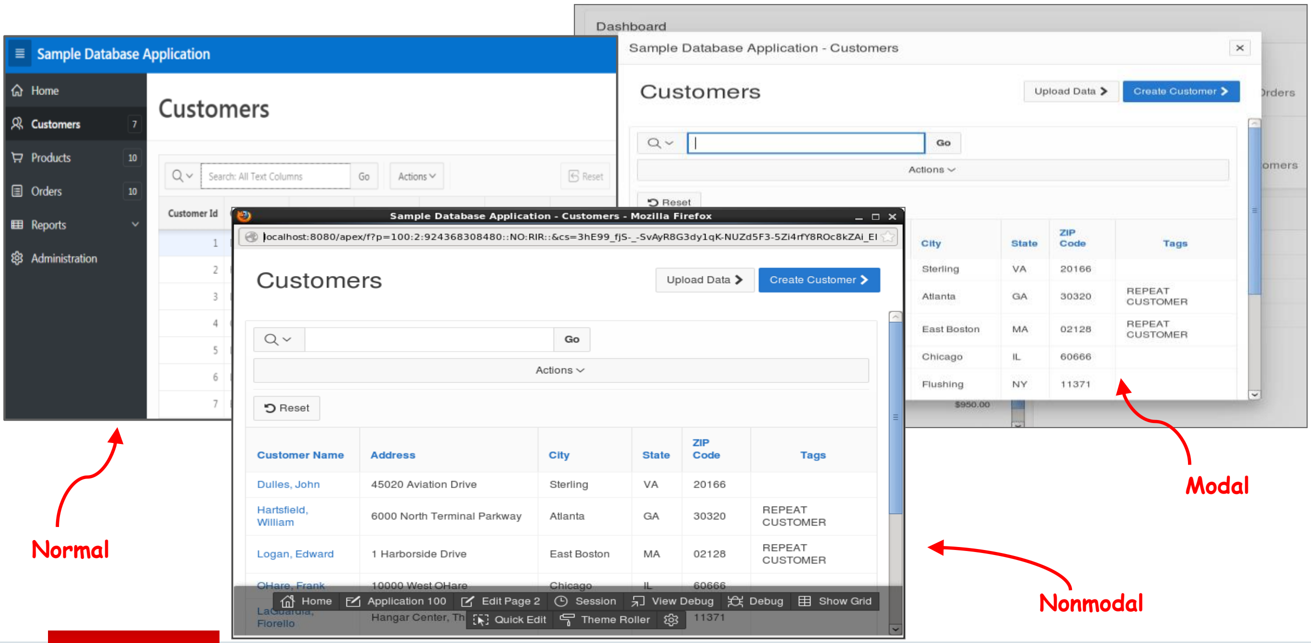
Name: Customers
Page Alias:
Title: &APP_NAME. - Customers
Page Group: Desktop
User Interface: Desktop
Page Mode: Normal
Page Template: Theme Default
Template Options: Use Template Defaults
CSS Classes:
Media Type:

Page Definition of the Customers page

So, why and how would you use a *Page Definition*? You use the *Page Definition* to view, create, and edit the components of a page. To access the page definition for a page, perform the following steps:

1. On the Oracle Application Express home page, click the **App Builder** icon.
2. On the App Builder page, click the application that you want to access.
3. On the selected application page, click a page to view its definition. Here, Steve has selected the *Customers* page. Remember, we learned that each page has a unique ID. Here 2 is the unique ID for the *Customers* page.
4. The page definition is displayed.

Page Modes: Normal, Modal, and Nonmodal


Sample Database Application
Home
Customers 7
Products 10
Orders 10
Reports
Administration
Customers
Dashboard
Sample Database Application - Customers
Upload Data
Create Customer
Go
Actions
Reset
Sample Database Application - Customers - Mozilla Firefox
localhost:8080/apex/f?p=100:2:924368308480::NO:RIR::&cs=3hE99_fjS-_-SvAyR8G3dy1qK-NUZd5F3-5Zi4rfY8ROc8kZAi_El
Home
Application 100
Edit Page 2
Session
View Debug
Debug
Show Grid
Quick Edit
Theme Roller
Normal
Modal
Nonmodal

Customer Name	Address	City	State	ZIP Code	Tags
Dulles, John	45020 Aviation Drive	Sterling	VA	20166	
Hartsfield, William	6000 North Terminal Parkway	Atlanta	GA	30320	REPEAT CUSTOMER
Logan, Edward	1 Harborside Drive	East Boston	MA	02128	REPEAT CUSTOMER
OHare, Frank	10000 West OHare	Chicago	IL	60666	
LaGuardia, Fiorello	Hangar Center, Th			11371	

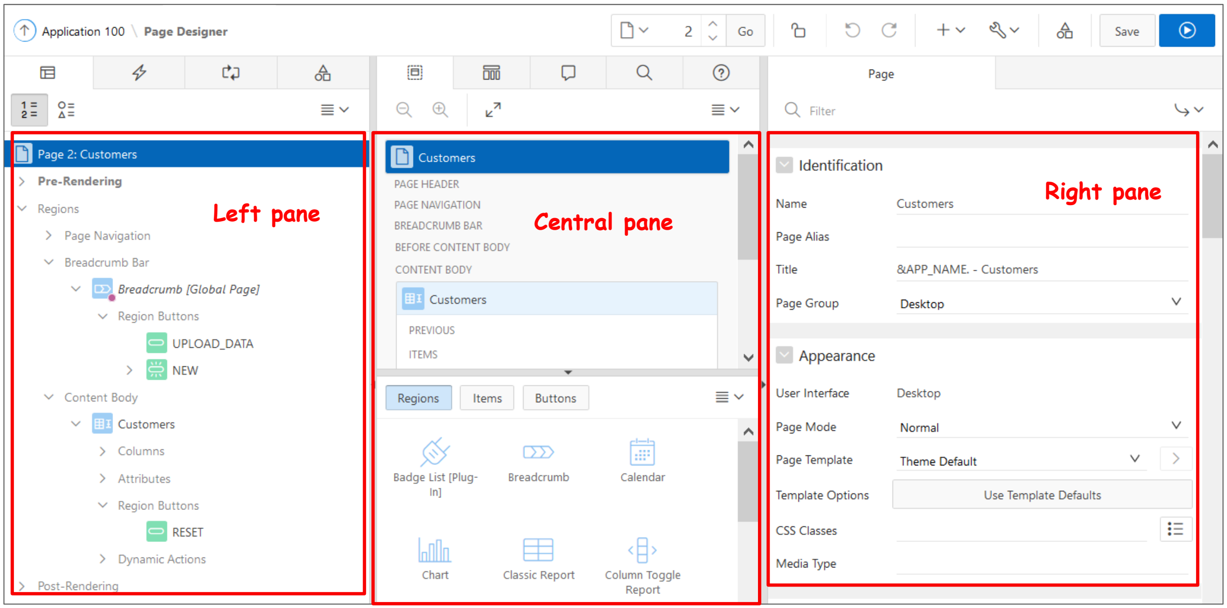
In the example in the slide, you can see that the same Customers page is displayed in three different ways. Oracle Application Express has this feature where you can choose the page display mode that you want for your application. Steve makes use of this functionality to enhance his UI and tries to give his pages a different look and feel.

As you see in the slide, Steve displays the *Customers* page of the *Sample Database Application* through the different page mode options (Normal, Modal Dialog, and Nonmodal Dialog) available in Oracle Application Express.

- **Normal:** This option displays the page as a normal Application Express application page.
- **Modal Dialog:** A modal dialog is an overlay window position within the same browser window. A modal dialog remains active and focused until you finish or close it. While a modal dialog is active, you cannot interact with the rest of the page until the dialog box is closed.
- **Nonmodal Dialog:** A nonmodal dialog displays a separate popup browser window. You can interact with a nonmodal dialog and content of the page and can use a nonmodal dialog when the requested information is not essential to continue. You can leave this type of window open while you continue work elsewhere.

You will be choosing your preferred display mode when you are working in the *Page Designer* window.

Page Designer Window



Identification

Name: Customers

Page Alias:

Title: &APP_NAME. - Customers

Page Group: Desktop

Appearance

User Interface: Desktop

Page Mode: Normal

Page Template: Theme Default

Template Options: Use Template Defaults

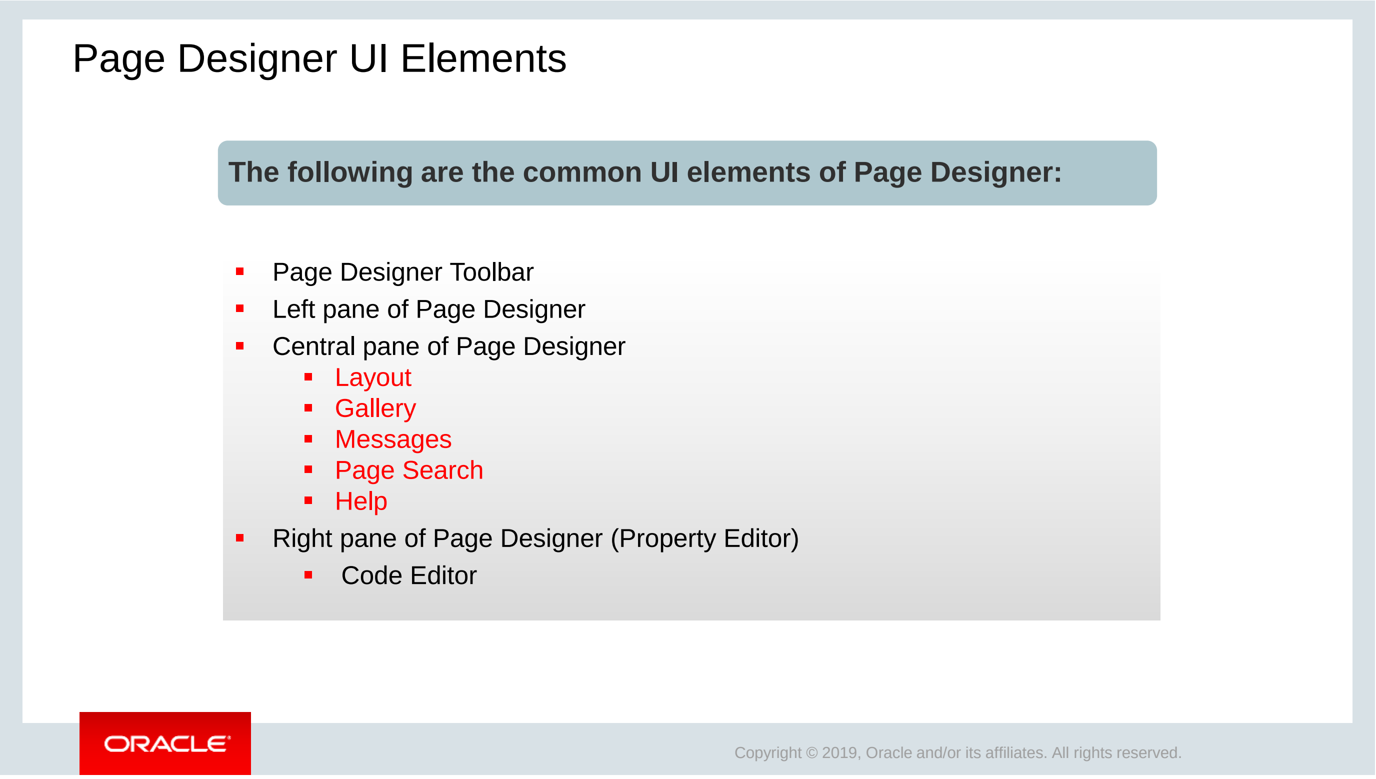
CSS Classes:

Media Type:

A *Page Designer* is a fully featured web-based Integrated Development Environment (IDE), which is designed to maximize your productivity for maintaining the application pages. It is very advanced and simple to use. Double-clicking any page from your application will open that particular page in Page Designer. In this slide, we have opened *Page 2: Customers* in the Page Designer.

As shown in the slide, the Page Designer window is divided into three panes:

- Left pane
- Central pane
- Right pane



Page Designer UI Elements

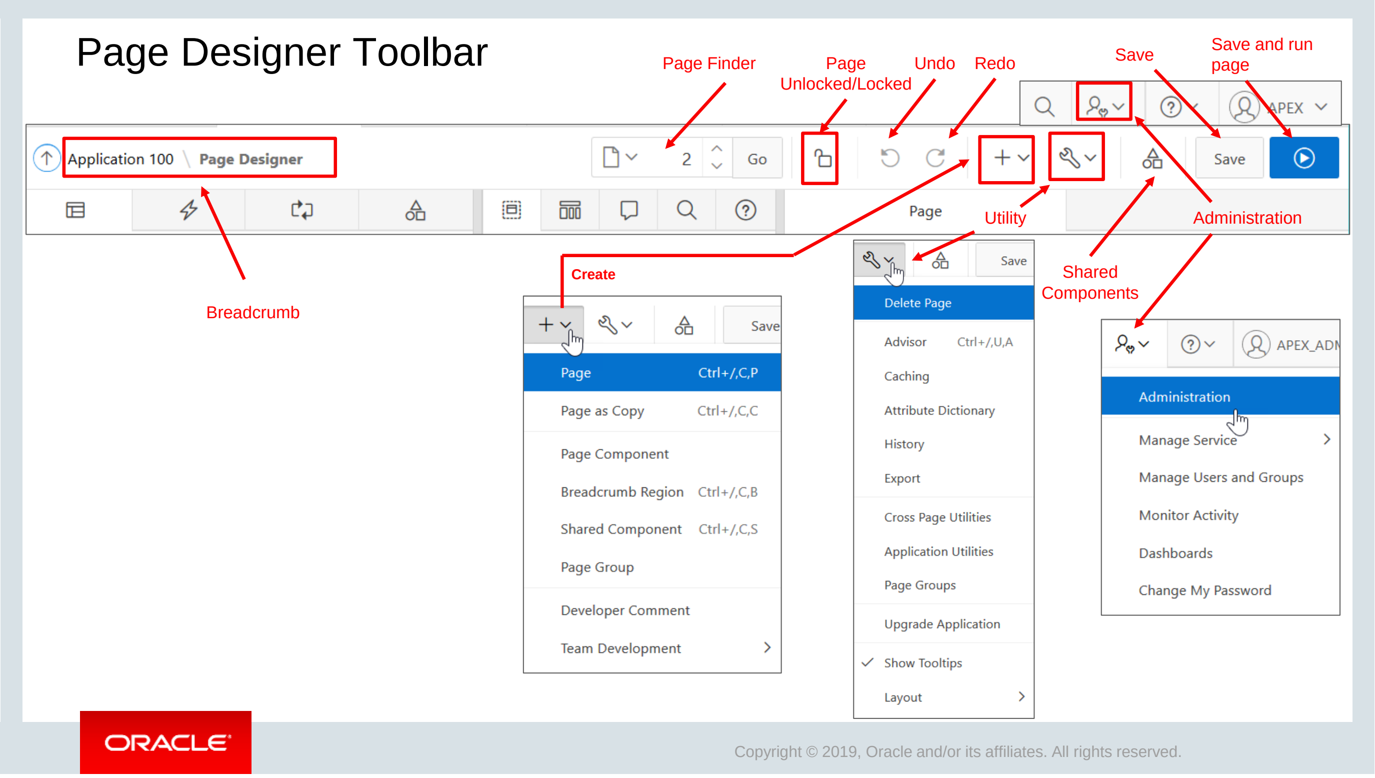
The following are the common UI elements of Page Designer:

- Page Designer Toolbar
- Left pane of Page Designer
- Central pane of Page Designer
  - Layout
  - Gallery
  - Messages
  - Page Search
  - Help
- Right pane of Page Designer (Property Editor)
  - Code Editor

ORACLE

The UI elements shown in the slide are explained in detail in the following slides.

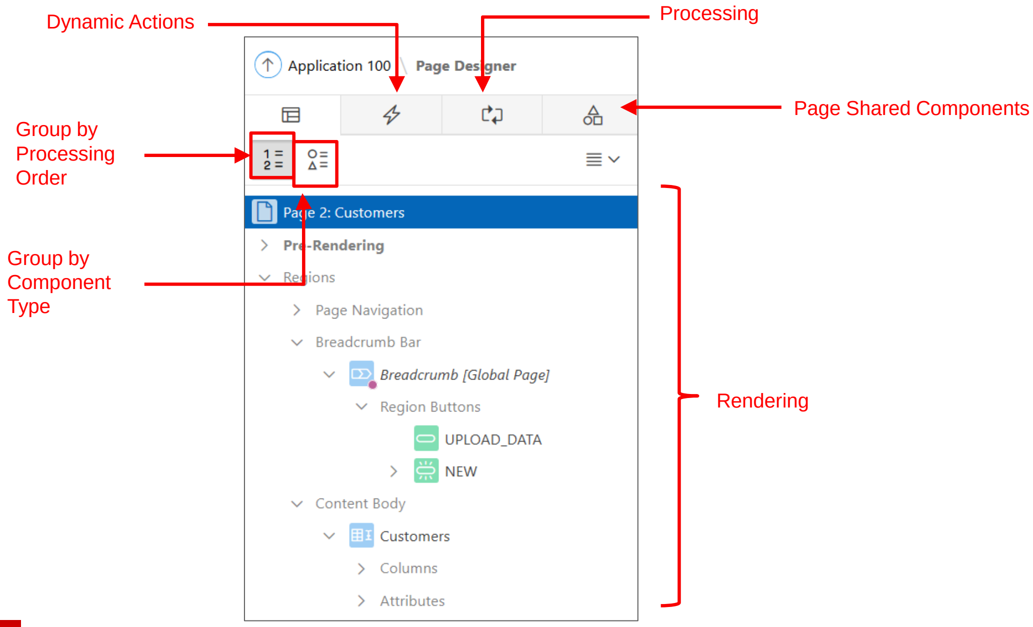
You will learn how to use these elements when you are editing the pages later in this lesson or creating and editing new pages for your application in later lessons of this unit.



Page Designer Toolbar

The Page Designer toolbar is displayed at the top of the Page Designer. Whenever you move your cursor over any item, button, or textbox, a tooltip is displayed. The Page Designer toolbar features multiple buttons and menu options that you will use frequently as you are working with the pages. See *Page Designer Toolbar* in Oracle Application Express documentation (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/page-designer-toolbar.html#GUID-5E911CA1-4A87-40B4-AAB7-57387D576DA0).

The Left Pane of Page Designer



This slide provides an overview of the Rendering Tree pane displayed on the left. It displays regions, page items, buttons, application logic (such as computations, processes, and validations), dynamic actions, branches, and shared components as nodes on a tree. The Tree pane features four tabs:

- **Rendering:** Displays regions, page items, page buttons, page components, and application logic as nodes in a tree
- **Dynamic Actions:** Displays dynamic actions defined on this page. You will learn how to create Dynamic Actions in the latter part of this unit.
- **Processing:** Displays application logic defined on this page. You will learn how to page processing works in the later part of this unit.
- **Page Shared Components:** Displays shared components associated with this page

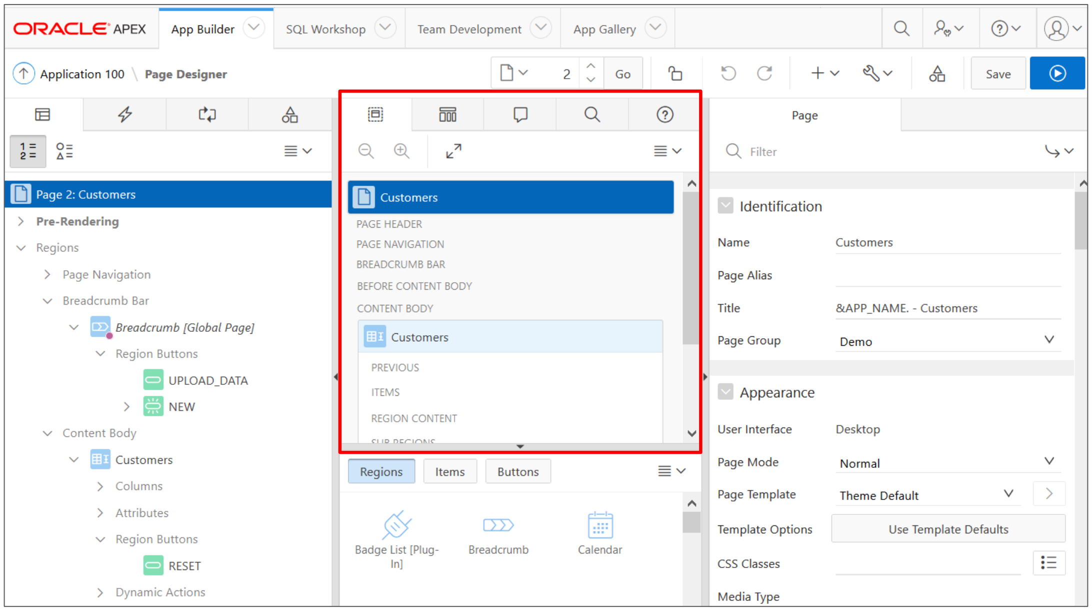
The Dynamic Actions and Page Shared Components trees are grouped by event or component type and then ordered by sequence. For example, if you click the Page Shared Components tab on the *Customers* page, you see that it is organized by Navigation Bar, Navigation Menu, Lists, Breadcrumbs, and so on.

The Rendering and Processing trees are grouped and ordered based on how Oracle Application Express processes them. This organization enables you to better understand when a component is processed and how the page is laid out. The Rendering and Processing trees have an option to group the nodes by component type. To control how tree components are displayed, click the **Group by Processing Order** or **Group by Component Type** buttons in the tree header bar.

Key features of the Tree pane include **Context menus, Quick Access to the Property Editor** (you can select a component or multiple components to display the corresponding attributes in the Property Editor in the right pane), **Drag and Drop**, and **Tooltips.**

For details, see *Left Pane of Page Designer* in the Oracle Application Express documentation (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/left-pane-page-designer.html#GUID-BF0FBF51-90D1-49E4-A269-DA7C16442179).

The Central Pane of Page Designer

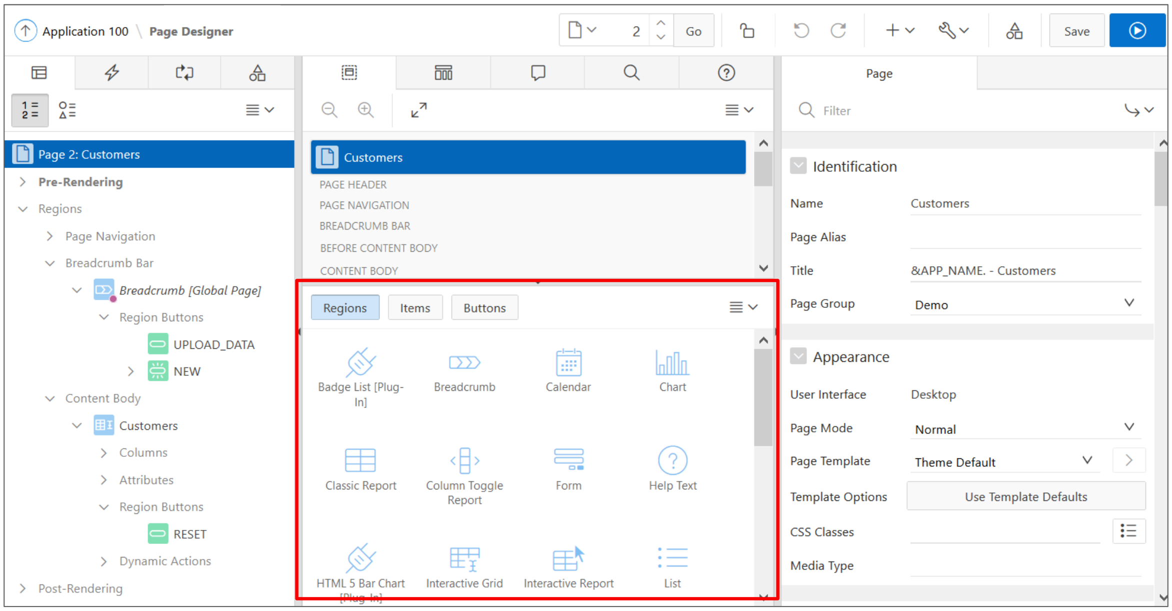


Name: Customers
Page Alias:
Title: &APP_NAME. - Customers
Page Group: Demo
User Interface: Desktop
Page Mode: Normal
Page Template: Theme Default
Template Options: Use Template Defaults
CSS Classes:
Media Type

In this slide, you focus on the central pane of the Page Designer. It contains five tabs: *Layout*, *Component View*, *Messages*, *Page Search*, and *Help*. The *Gallery*, which is also a part of the central pane, appears below. You learn that in detail in the next slide. You also learn how to use these tabs when you are editing a page later in this lesson.

For details, see *Central Pane of Page Designer* in the Oracle Application Express documentation (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/central-pane-page-designer.html#GUID-06BF3282-A0D3-4261-90DA-1A96BCB37527).

The Central Pane of Page Designer: Gallery



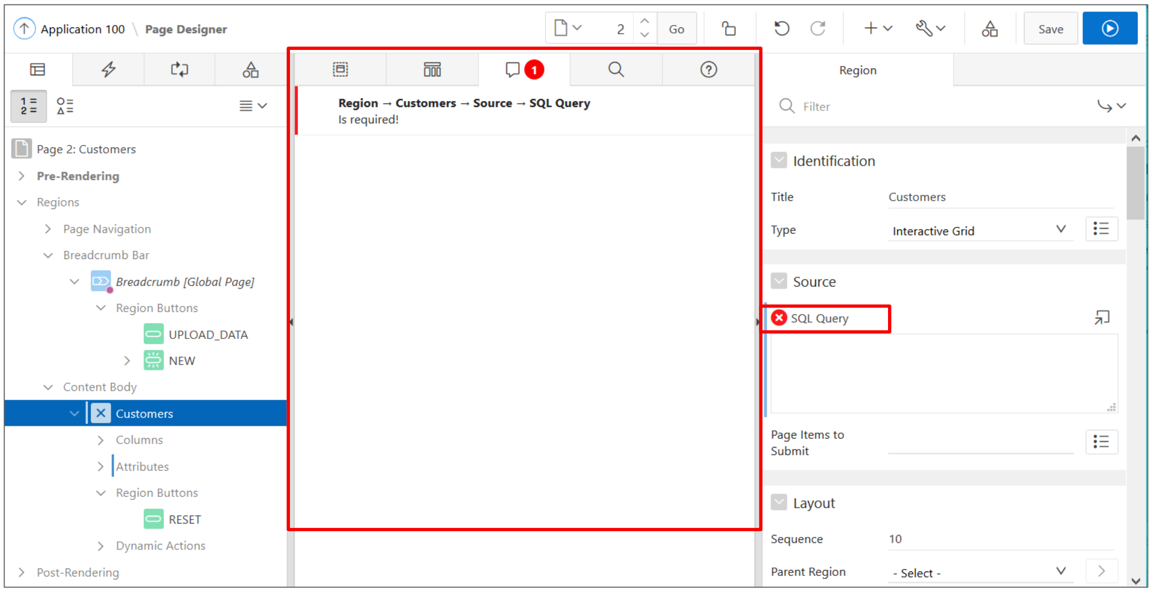
Name	Customers
Page Alias	
Title	&APP_NAME. - Customers
Page Group	Demo
User Interface	Desktop
Page Mode	Normal
Page Template	Theme Default
Template Options	Use Template Defaults
CSS Classes	
Media Type	

The *Gallery* is displayed at the bottom of the *Layout* pane and features three tabs: *Regions*, *Items*, and *Buttons*. By default, only supported controls and components appear. When you place the cursor over a control or component, a tooltip is displayed that describes it. In this slide, you see the tooltip for a Classic Report.

You can add new controls and components to a page by selecting them in the *Gallery* and dragging them into *Layout*. Steve tries out this option in the next slide and creates an Item from the Gallery pane.

For details, see *Gallery* in the Oracle Application Express documentation (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/central-pane-page-designer.html#GUID-E699A89C-336B-4CFC-8E9A-9EB7A7CD0CA3).

The Central Pane of Page Designer: Messages



Region → Customers → Source → SQL Query
Is required!

Title: Customers
Type: Interactive Grid
SQL Query
Sequence: 10

Steve has now created a List of Values item (discussed in a later chapter), but he forgot to enter the details in the *Type* field. As you can see in the slide, the *Messages* tab flashes a red badge, which indicates that there are some errors that need to be addressed.

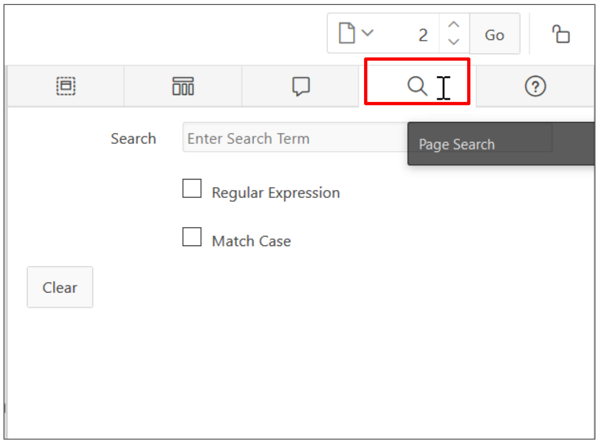
A great feature of Oracle Application Express is that it notifies you if you have made any mistakes while creating components or editing attributes in Page Designer.

The Messages tab displays a red or yellow badge indicating the number of messages you need to address. The Messages tab displays two types of messages:

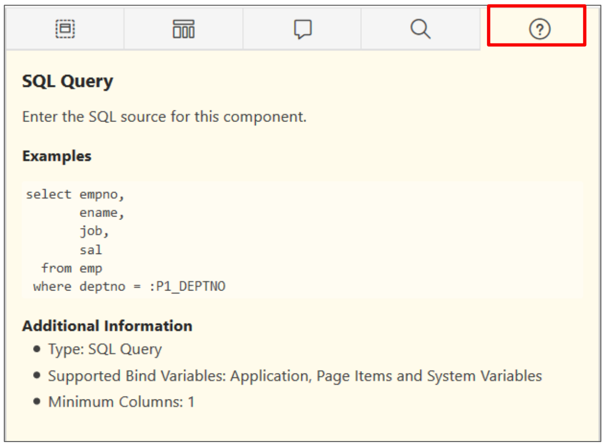
- **Errors:** Error messages are displayed in red. Selecting an error message displays the associated attribute in red in the Property Editor. You must address errors before a page can be saved.
- **Warnings:** Warning messages are displayed in yellow. Selecting a warning message displays the associated attribute in yellow in the Property Editor. You can save a page without addressing warning messages.

The Central Pane of Page Designer: Page Search and Help

Page Search



Help



SQL Query

Enter the SQL source for this component.

Examples

```
select empno,
       ename,
       job,
       sal
  from emp
 where deptno = :P1_DEPTNO
```

Additional Information

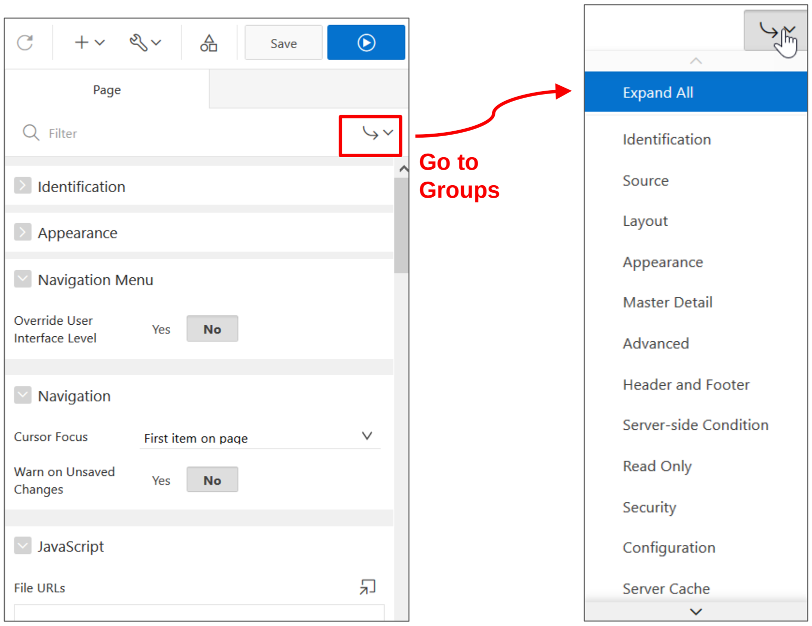
- Type: SQL Query
- Supported Bind Variables: Application, Page Items and System Variables
- Minimum Columns: 1

Page Search is another user-friendly feature of Oracle Application Express. As you see in the slide, it appears to the right of Messages in the central pane. You can use *Page Search* to search all page metadata including regions, items, buttons, dynamic actions, columns, and so on.

- To search a page, you need to enter a search term in the field provided.
- To match the case, select **Match Case**.
- To search for a regular expression, select **Regular Expression**.

Page Designer also includes *Help* for every Property Editor attribute. To view Help, select the attribute and click the *Help* tab. After you activate the Help pane, the content that is displayed changes every time you select another attribute.

The Right Pane of Page Designer: Property Editor



Override User Interface Level: Yes **No**

Cursor Focus: First item on page

Warn on Unsaved Changes: Yes **No**

File URLs

Go to Groups

Now that you have learned about the left and central panes of the Page Designer, let's see what we have in the right pane. As you see, the Property Editor is displayed in the right pane, and it displays all attributes for the current component.

As you select different components in either Tree View or Layout, the Property Editor automatically updates to reflect the current selection. Attributes are organized into functional groups that describe their purpose. You can control the amount of information that is displayed using the multiple options under **Go to Group.**

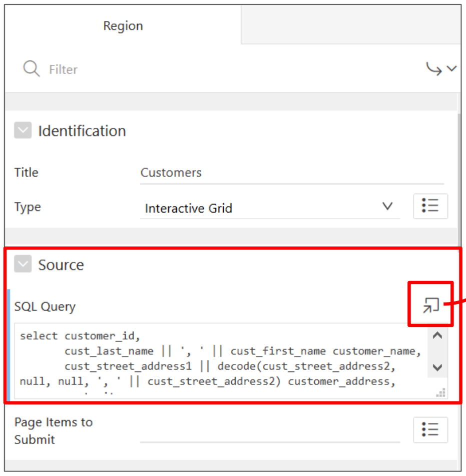
The Property Editor organizes attributes into functional groups. To quickly access a group, click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

You will be using the Property Editor extensively in the upcoming slides and lessons when you are creating a new component or updating an existing component.

For details, see *Right Pane of Page Designer (Property Editor)* in the Oracle Application Express documentation (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/right-pane-page-designer.html#GUID-CDCB9B74-8BE1-4ABE-93AB-099AEF3742BD).

Page Designer UI Elements: Property Editor: Code Editor

Enter text directly into the field.



Identification

Title: Customers

Type: Interactive Grid

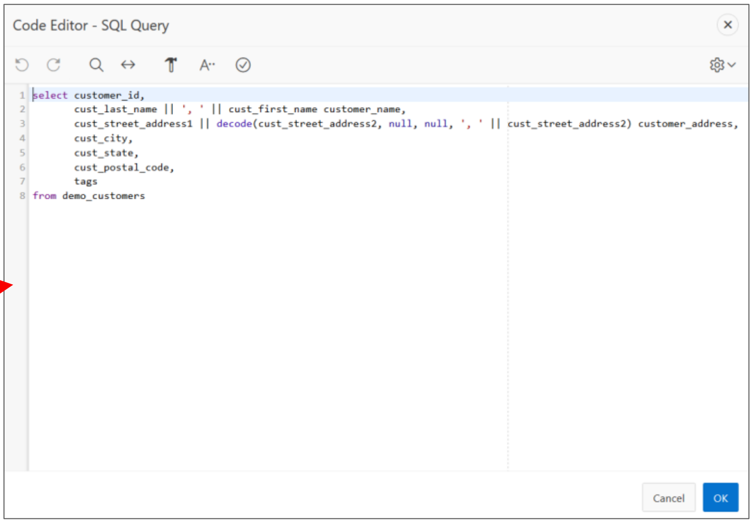
Source

SQL Query

```
select customer_id,
       cust_last_name || ', ' || cust_first_name customer_name,
       cust_street_address1 || decode(cust_street_address2,
null, null, ', ' || cust_street_address2) customer_address,
```

Page Items to Submit

Enter text into the Code Editor.



Code Editor - SQL Query

```
select customer_id,
       cust_last_name || ', ' || cust_first_name customer_name,
       cust_street_address1 || decode(cust_street_address2, null, null, ', ' || cust_street_address2) customer_address,
       cust_city,
       cust_state,
       cust_postal_code,
       tags
from demo_customers
```

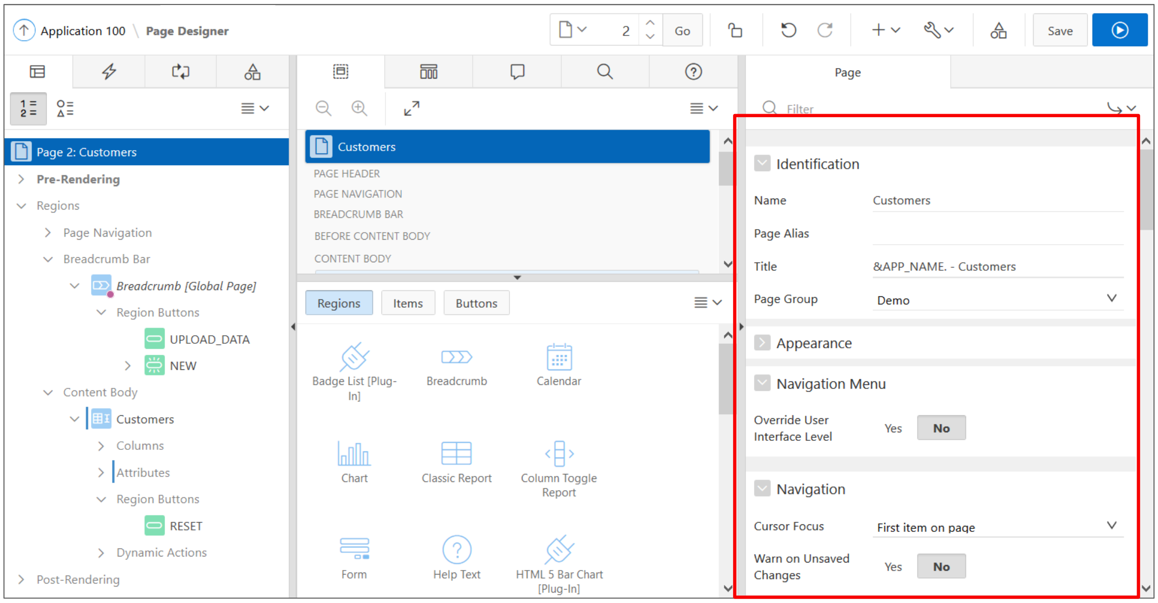
The Code Editor appears under the *Source* attribute in the Property Editor. For attributes requiring large amounts of code (for example, a Region Source), you can either:

- Enter text directly into the field (as you see in the image on the left in the slide)
- Click **Code Editor** to enter text into a full featured code editor

A Code Editor is displayed as a modal dialog. (The different page modes were discussed earlier in this lesson.) You use the Code Editor to edit the attributes that require a large amount of code. The Code Editor provides an enhanced code editor to edit PL/SQL, SQL, HTML, CSS, and JavaScript component properties. The code highlighting is determined by the input required for the specific property.

Key features of the Code Editor include Undo, Redo, Find, Replace, Query Builder, Autocomplete, Validate, Syntax highlighting, Block indent and unindent, and Resize dialog box.

Editing Page Attributes



Identification

Name: Customers

Page Alias:

Title: &APP_NAME. - Customers

Page Group: Demo

Appearance

Navigation Menu

Override User Interface Level: Yes [✓ No]

Navigation

Cursor Focus: First item on page

Warn on Unsaved Changes: Yes [✓ No]

Now that you are comfortable with the different tabs and attributes in the Page Designer, let’s see how you can edit the attributes of a page in your application.

To edit page attributes, perform the following steps:

1. View the page definition in Page Designer.
2. In either the Rendering Tree or Layout, select the page name. The page name is selected by default when you access the Page Definition. The Property Editor displays the page attributes.
3. Edit the appropriate attributes in the Property Editor.

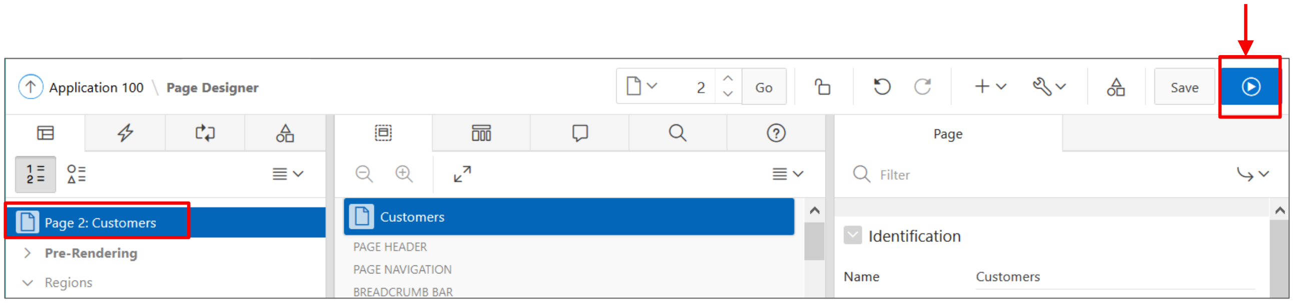
A few attributes of a page are listed below:

- **Name:** It identifies the name of the current page for application developers. This name is used in numerous pages and reports, along with the page number and page title.
- **Page Alias:** Enter an alphanumeric alias for this page. This alias must be unique with the current application. For example, if you were working on page 1 of application 100, you could create an alias called home. You could then access this page from other pages using the following f?p syntax: `f?p=100:home`
- **Title:** Enter a title to be displayed in the title bar of the browser window.
- **Page Mode:** Select a page mode (Normal, Modal Dialog, and Nonmodal Dialog).

To learn more about an attribute, use the *Help* tab in the center pane.

Running a Page

Save and Run Page



- View the page definition in Page Designer.
- Click the “Save and Run Page” icon.

After you have created or updated the page, you would like to see the changed version of your application. The Application Express engine dynamically renders and processes pages based on data stored in database tables. To view a rendered version of an application, you run or submit it to the Application Express engine.

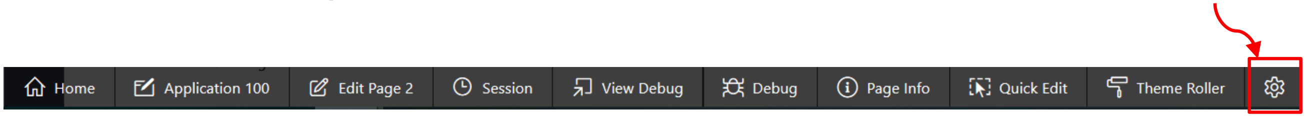
To run a page:

- View the page definition in Page Designer.
- Click the **Save and Run Page** icon in the upper-right corner.

Note: You cannot run Modal and Nonmodal dialog pages directly using the “Save and Run Page” icon. To view a Modal or Nonmodal dialog page, run the normal page that is responsible for opening the dialog page and access it from there.

Runtime Developer Toolbar

Developer toolbar options



- **Home:** Links to the Workspace home page
- **Application:** Links to the Application home page
- **Edit Page:** Opens the current page in edit mode
- **Session:** Opens the session state information
- **View Debug:** Displays the Debug reports
- **Debug:** Toggles the page between **Debug** and **No Debug** mode
- **Page Info:** Shows Layout Columns and Page Timing
- **Quick Edit:** Enables developers to enter either Quick Edit mode or access Live Template Options
- **Theme Roller:** Enables developers to easily customize the appearance of an application

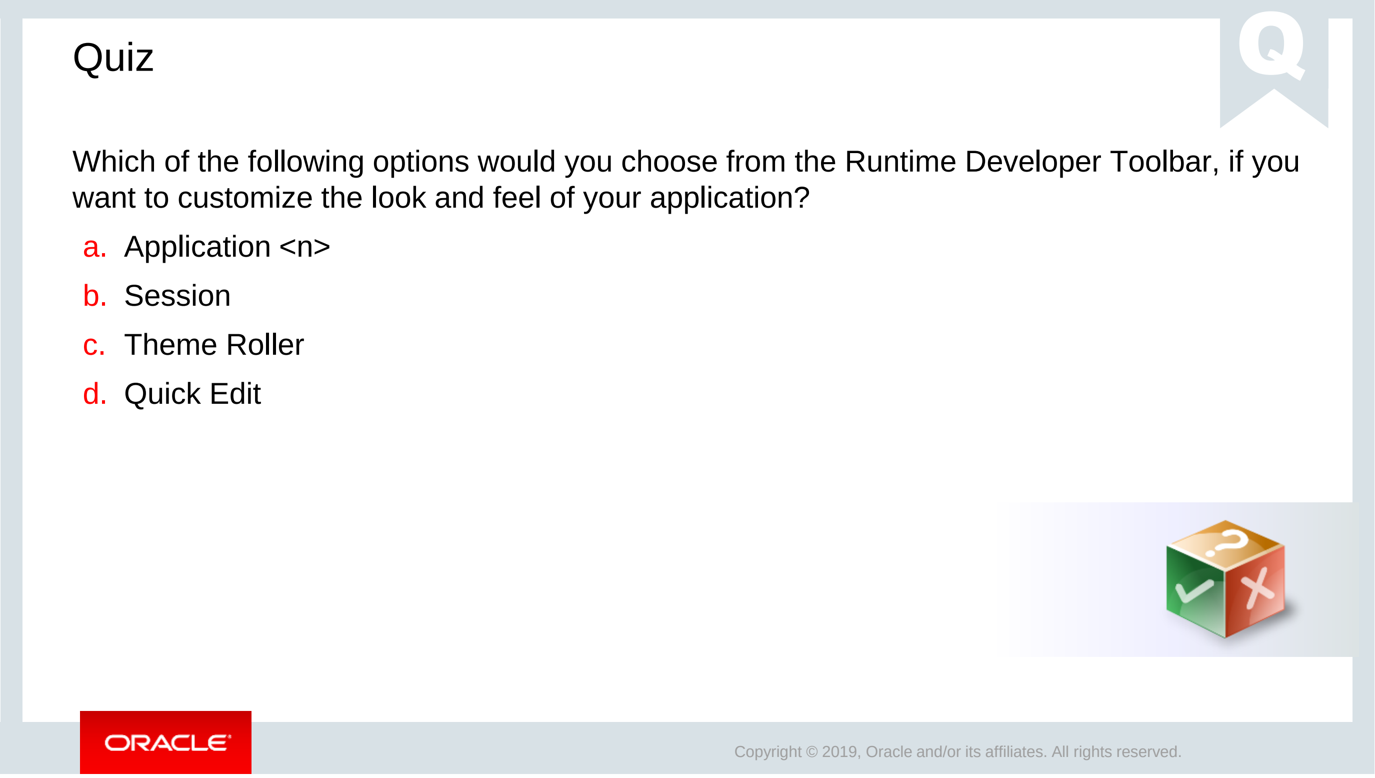
ORACLE

When you run an application from Application Builder, you can see the Runtime Developer toolbar displayed at the bottom of any editable running page. You can use this Runtime Developer toolbar to quickly edit the current application or currently running page or view session state.

If a page has one or more JavaScript errors, a red error button displays on the left side of the Runtime Developer toolbar. If you have enabled **Auto Hide**, the Runtime Developer toolbar is displayed indicating the error.

Notes

- **Page Info:** Contains the **Show Layout Columns** and **Show Page Timing** features
  - **Show Layout Columns** toggles between **Show Layout Columns** and **Hide Layout Columns**.
  - **Show Page Timing** displays the Page Performance Timing dialog box. Click **Copy** to copy the data in table form and then paste it into another application. Click **Clear** to remove the current timing events.
- **Quick Edit:** Enables developers to enter either Quick Edit mode or access Live Template Options.
  - Access Quick Edit Mode - Click **Quick Edit** and then select the desired component to instantly access the component in Page Designer. Press ESCAPE or click outside a component to exit quick edit mode.
  - Edit Live Template Options - Click **Quick Edit** and then move the mouse over the component for which you want to modify template options and click the Wrench icon in the upper-right corner.
- **Theme Roller:** Enables developers to easily customize the appearance of an application by selecting colors from color picker and setting values. It is displayed only for themes that support Theme Roller.
- **Developer Toolbar Options:** Displays on the far right of the Developer toolbar. You can customize how the Developer toolbar is displayed by using the following controls:
  - **Auto Hide**
  - **Show Icons Only**
  - **Display Position** (Top, Left, Bottom, Right)



Quiz

Which of the following options would you choose from the Runtime Developer Toolbar, if you want to customize the look and feel of your application?

a. Application <n>

b. Session

c. Theme Roller

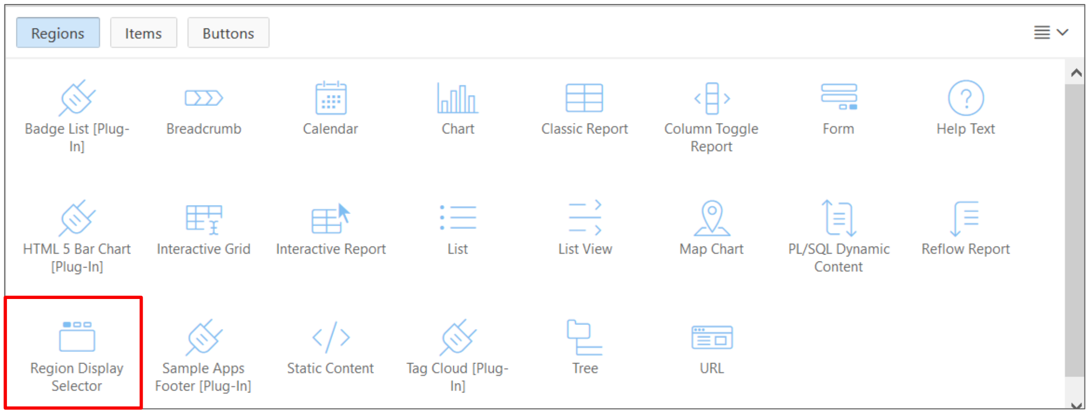
d. Quick Edit

Answer: c

Lesson Agenda

- Introducing Page Definition
- Working with Page Regions
  - Understanding Region and Region Types
  - Adding a Region
  - Conditional Display of Regions
  - Positioning a Region
  - Creating a Region Display Selector
  - Viewing and Editing Region Attributes
  - Specifying Region Header or Footer
  - Copying Regions
  - Creating a Subregion
- Working with Pages

Region and Region Types



In the previous slides, you learned about the left, central, and right panes in the Page Designer. Remember you saw this screenshot when we were discussing *Gallery* region under the Central pane. Now, in the next few slides, you will learn about regions, types of regions, and how to add regions.

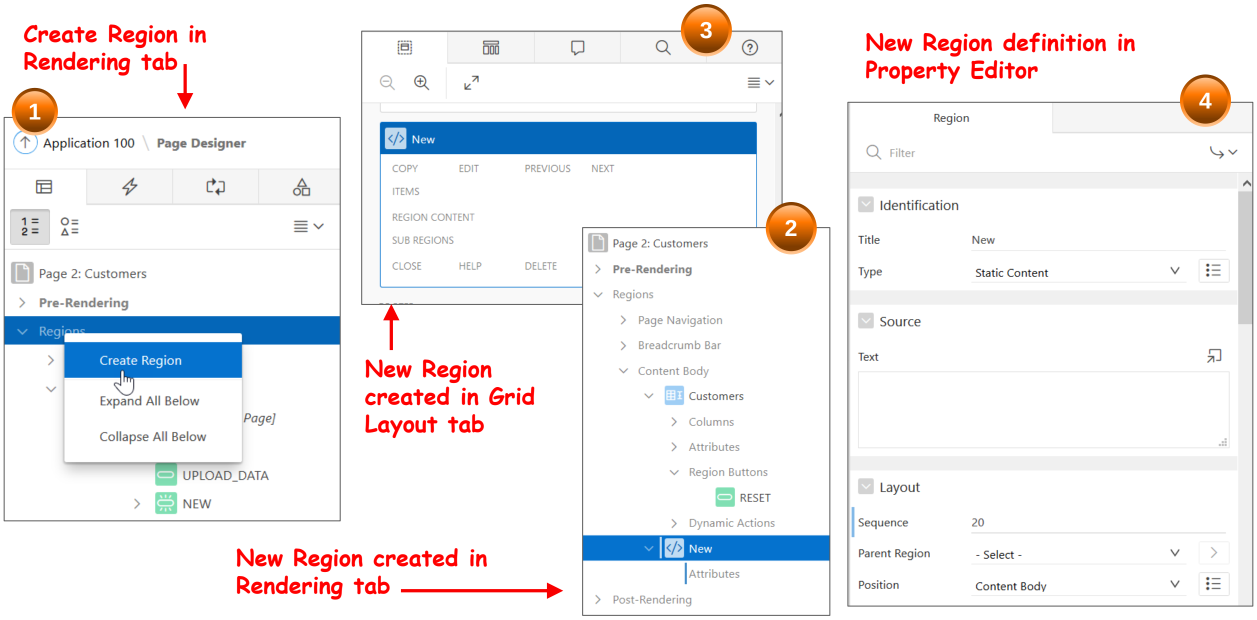
Regions are basically containers. Each page can have buttons and fields (called items), which are grouped into containers called regions.

You can add regions to a page from the *Rendering Tree* or from the *Gallery*. You will see how to add regions using both these options in the next slides.

The slide above shows you the different region types available in Oracle Application Express. When you create a region, you also select a region type. The Application Express engine interprets a region differently based on the region type you select. You will learn how to create **Region Display Selector** region type later in this lesson.

Also, as you see in the screenshot in this slide, there are two other tabs, *Items* and *Buttons*, which will be covered in a later lesson.

Adding a Region: From the Rendering Tree



Create Region in Rendering tab

New Region created in Grid Layout tab

New Region definition in Property Editor

New Region created in Rendering tab

Title: New

Type: Static Content

Sequence: 20

Parent Region: - Select -

Position: Content Body

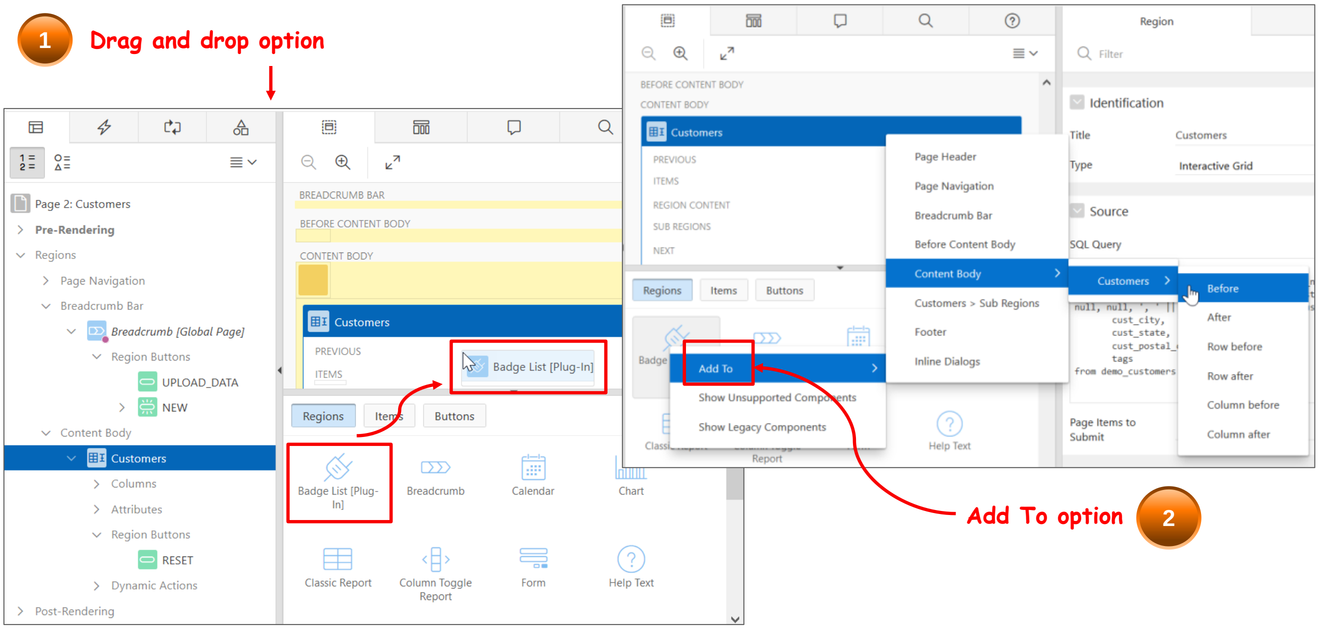
To add a region from a Rendering Tree, perform the following steps:

1. View the page in Page Designer.
2. On the Rendering tab in the left pane, right-click **Regions** and select **Create Region** (screenshot 1).
3. The Region is created (screenshots 2 and 3). Edit the appropriate attributes in the Property Editor (screenshot 4).
4. Click **Save** or **Save and Run Page**.

Note that when you right-click in an existing region for example, here the Customers, you get the options to Create a Region, Create a Sub Region, Create a Button, Create a Page Item, Delete, Duplicate, etc.

Adding a Region: From the Gallery

1 Drag and drop option

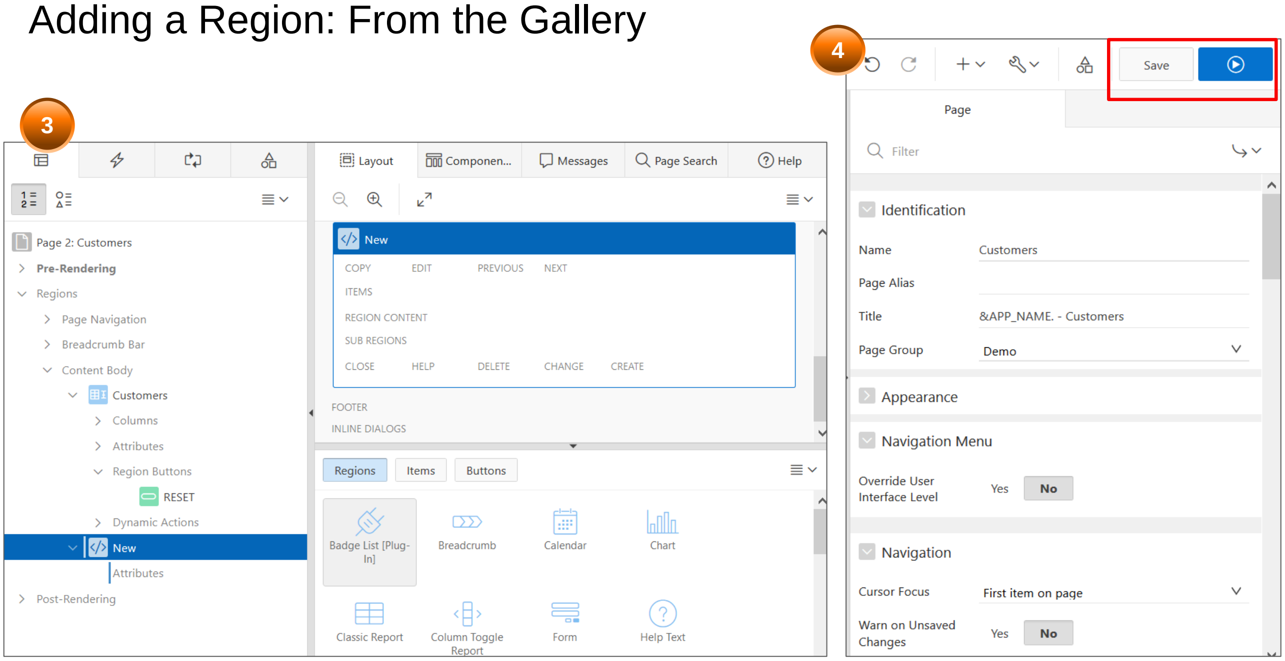


Add To option 2

To add a region from the *Gallery*, perform the following steps:

1. View the page in Page Designer.
2. In the Gallery, locate the region you want to add.
    - Click and hold the mouse the component to be created and drag it to the desired location in Grid Layout (screenshot 1).
    - Or right-click a component to view a context menu. For **Add To**, select the location where you want the region to be added (screenshot 2).

Adding a Region: From the Gallery



Name: Customers

Page Alias:

Title: &APP_NAME. - Customers

Page Group: Demo

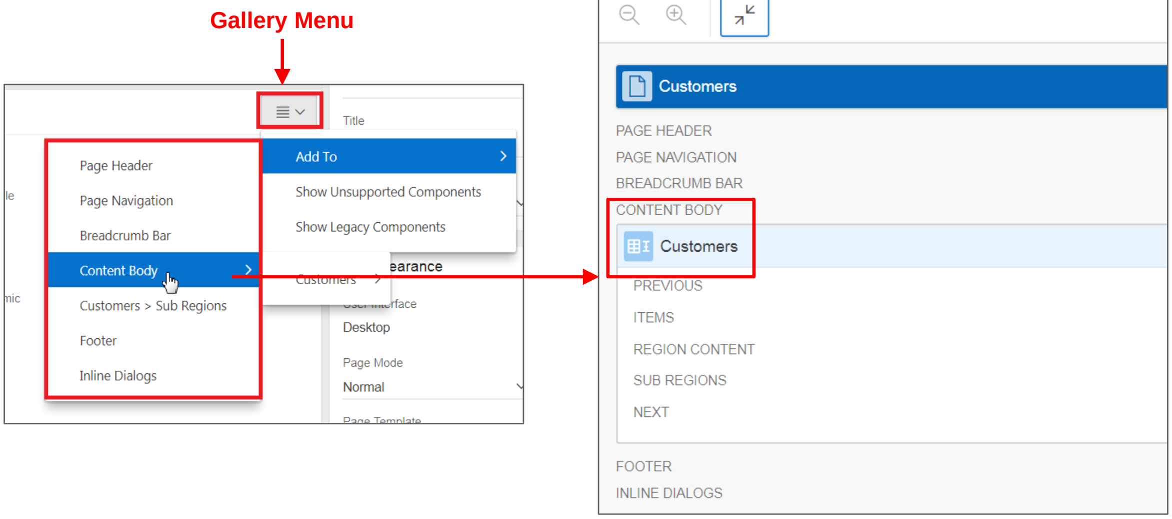
Override User Interface Level: Yes [✓ No]

Cursor Focus: First item on page

Warn on Unsaved Changes: Yes [✓ No]

3. The Region is created (screenshot 3). Edit the appropriate attributes in the Property Editor (screenshot 4).
4. Click **Save** or **Save and Run Page**.

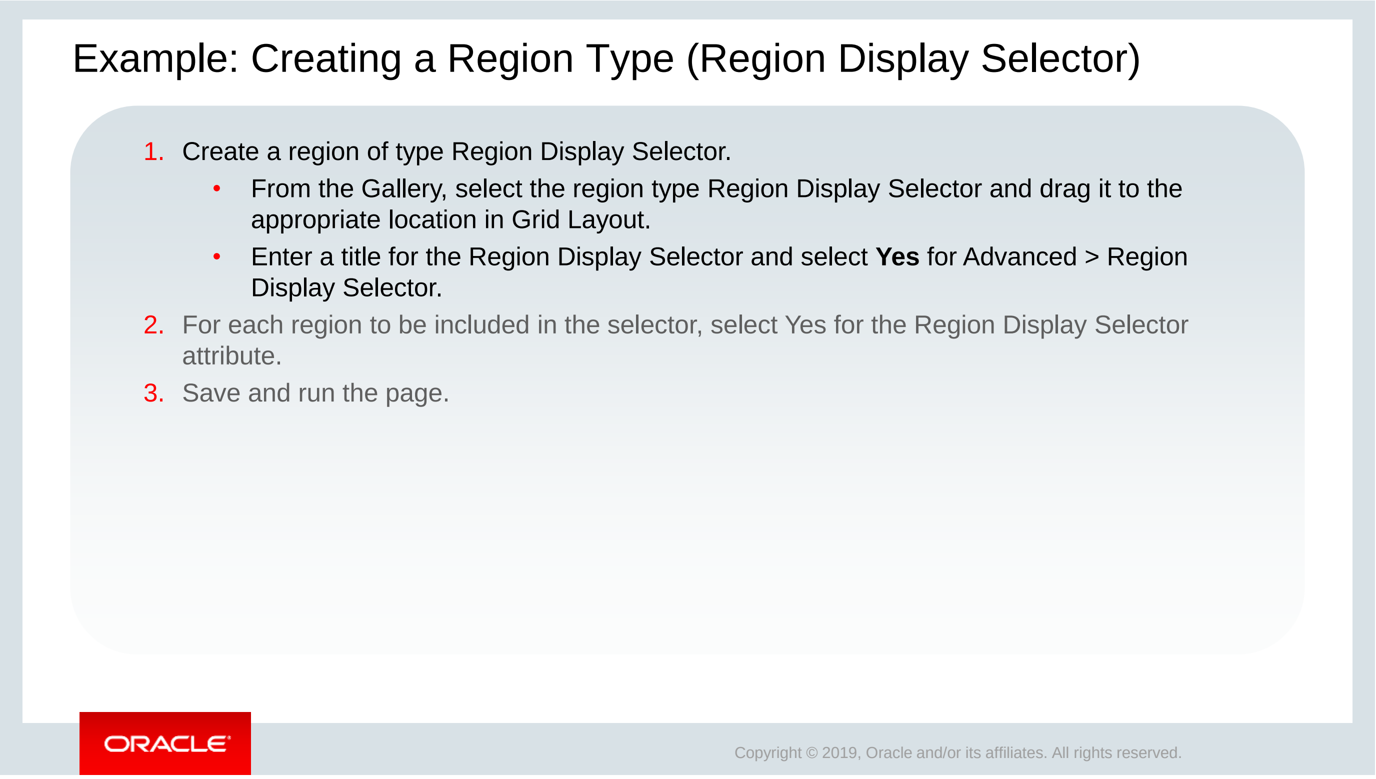
Positioning the Region



When you create a region, you must specify its **Position** on the page. Click the **Gallery Menu** icon to display the available options. You can select either a default position (such as Content Body) or a position as per your choice.

You can also specify the sequence of the region to position the region in relation to other regions on the page.

Note: The Parent Region field is used to create a subregion. You learn to create a subregion later in the lesson.

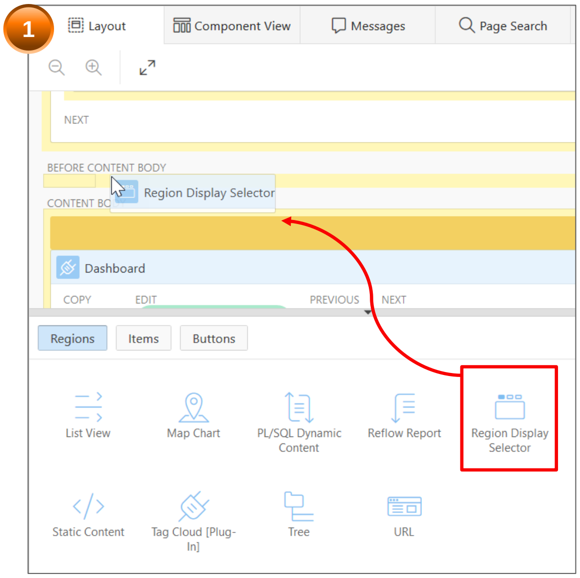


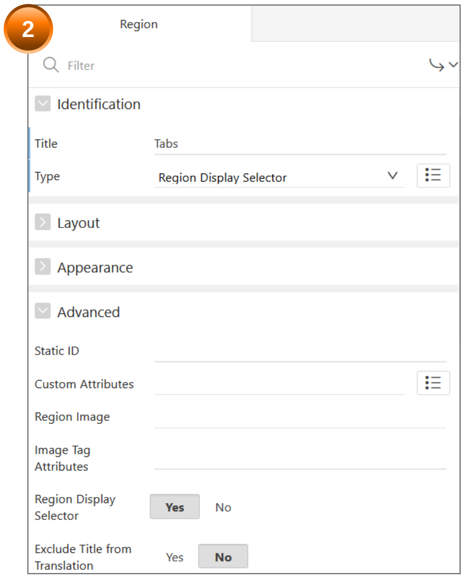
Example: Creating a Region Type (Region Display Selector)

1. Create a region of type Region Display Selector.
   - From the Gallery, select the region type Region Display Selector and drag it to the appropriate location in Grid Layout.
   - Enter a title for the Region Display Selector and select **Yes** for Advanced > Region Display Selector.
2. For each region to be included in the selector, select Yes for the Region Display Selector attribute.
3. Save and run the page.

In the section “Understanding Region and Region Types” earlier in this lesson, you learned about the different region types. This slide provides an overview of how you can create the region type called Region Display Selector. The **Region Display Selector** region enables you to include show and hide controls for each region on a page.

Example: Creating a Region Display Selector

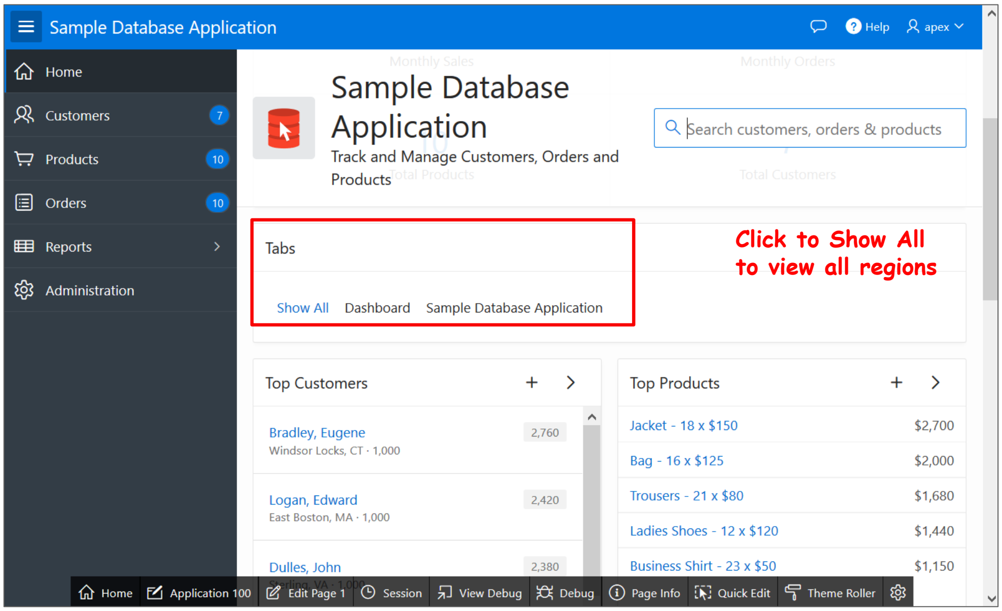




Title: Tabs
Type: Region Display Selector
Static ID:
Custom Attributes:
Region Image:
Image Tag Attributes:
Region Display Selector: [✓ Yes] No
Exclude Title from Translation: Yes [✓ No]

Steve creates this region type to display the list horizontally. This will enable the end user to select one region to display and hide the other region. In this example, the user can switch between *Show All*, *Dashboard*, and *Sample Database Application* tabs.

Example: Creating a Region Display Selector

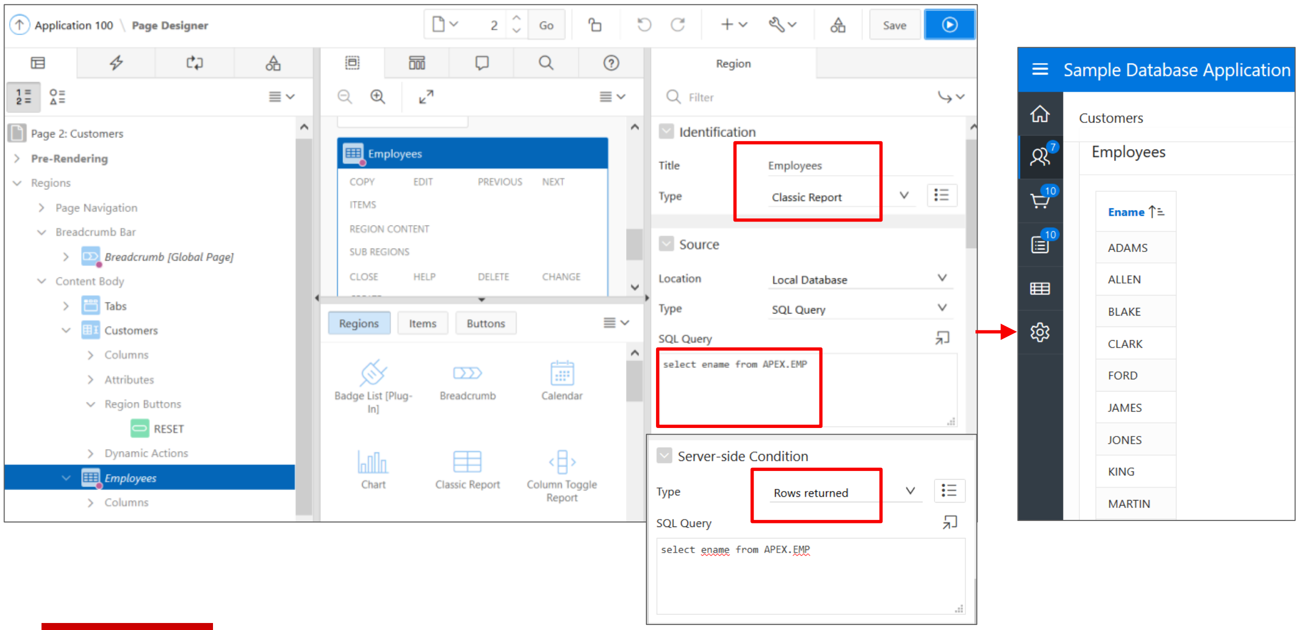


Sample Database Application
Help
apex
Home
Customers 7
Products 10
Orders 10
Reports
Administration
Sample Database Application
Track and Manage Customers, Orders and Products
Search customers, orders & products
Tabs
Show All
Dashboard
Sample Database Application
Top Customers
Bradley, Eugene 2,760
Windsor Locks, CT · 1,000
Logan, Edward 2,420
East Boston, MA · 1,000
Dulles, John 2,380
Top Products
Jacket - 18 x $150 $2,700
Bag - 16 x $125 $2,000
Trousers - 21 x $80 $1,680
Ladies Shoes - 12 x $120 $1,440
Business Shirt - 23 x $50 $1,150
Home
Application 100
Edit Page 1
Session
View Debug
Debug
Page Info
Quick Edit
Theme Roller

Click to Show All to view all regions

The screenshot shows the Sample Database Application with the *Tabs* Region Display Selector. In this example, the user can switch between *Show All*, *Dashboard*, and *Sample Database Application* tabs.

Conditional Display of Regions



Identification

Title: Employees

Type: Classic Report

Source

Location: Local Database

Type: SQL Query

SQL Query:

```
select ename from APEX.EMP
```

Server-side Condition

Type: Rows returned

SQL Query:

```
select ename from APEX.EMP
```

Ename
ADAMS
ALLEN
BLAKE
CLARK
FORD
JAMES
JONES
KING
MARTIN

ORACLE

You can also display regions conditionally. A condition is a small unit of logic that enables you to control the display of regions, items, buttons, tabs, and other components. When you apply the condition to a component (for example, a region), the condition is evaluated at run time. The component is displayed only if the condition evaluates to true.

You can set the condition by selecting a condition type when you create the component or by editing the component's conditional display attribute. For example, if you want a particular region to be displayed only when the administrator logs in, you can set an appropriate condition for that region. The condition evaluates to true or false based on the values you enter in the expression fields (SQL Query, Item/Value, or Expression). To set condition:

1. Either create a new region or select the region you want to attach the condition to. In this slide, you create a **New region**.
2. In the Property Editor, enter the following:
   - **Identification**: **Title**: Enter `Employees`.
   - **Identification**: **Type**: Select `Classic Report`.
   - **Source**: **Type**: Select `SQL Query`.
   - **Source**: **SQL Query** (code text box): Enter:

```
select ename
from
APEX.EMP
```

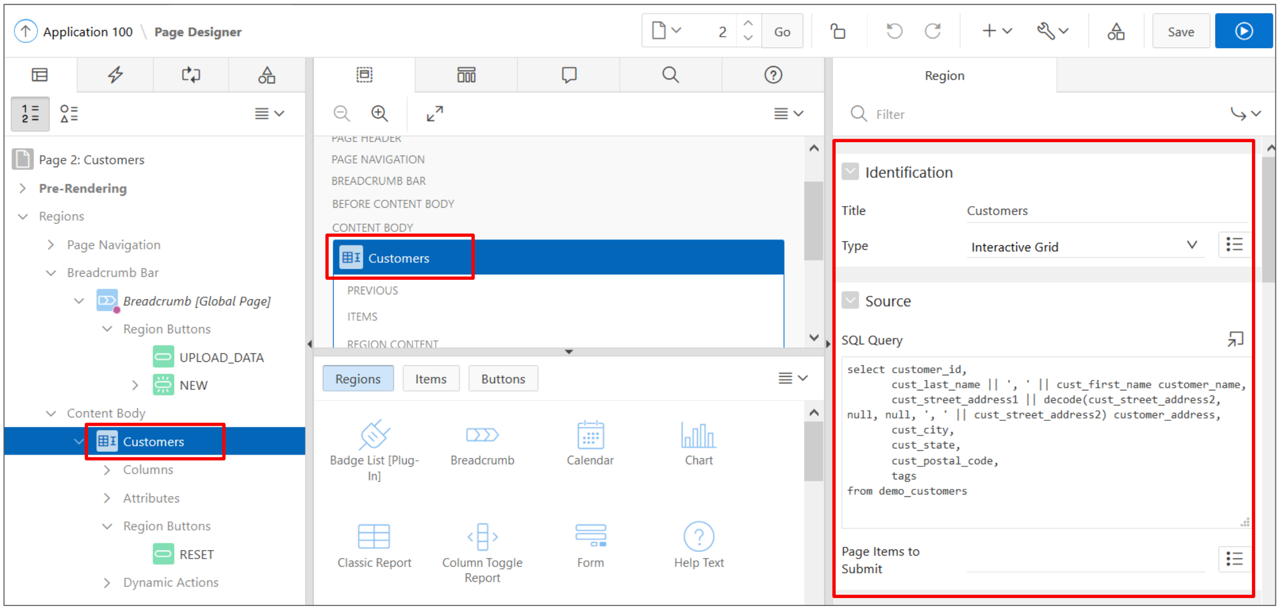
3. In the Property Editor, navigate to the Server-side Condition tab and select the appropriate Type from the drop-down icon. Here, you choose the following:
   - **Server-side Condition**: **Type**: `Rows Returned`
4. After selecting the condition type, fill in the SQL Query/Item/Value/Expression depending on the Type selected.
   - **Server-side Condition**: **SQL Query** (code text box): Enter:

   ```
   select ename

   from

   APEX.EMP
   ```
5. When you run the page, if the condition evaluates to true, the region is displayed. In this example, the Region is displayed only if the `EMP` table has records in it.

Editing a Region



Identification

Title: Customers

Type: Interactive Grid

Source

SQL Query

```
select customer_id,
       cust_last_name || ', ' || cust_first_name customer_name,
       cust_street_address1 || decode(cust_street_address2,
null, null, ', ' || cust_street_address2) customer_address,
       cust_city,
       cust_state,
       cust_postal_code,
       tags
from demo_customers
```

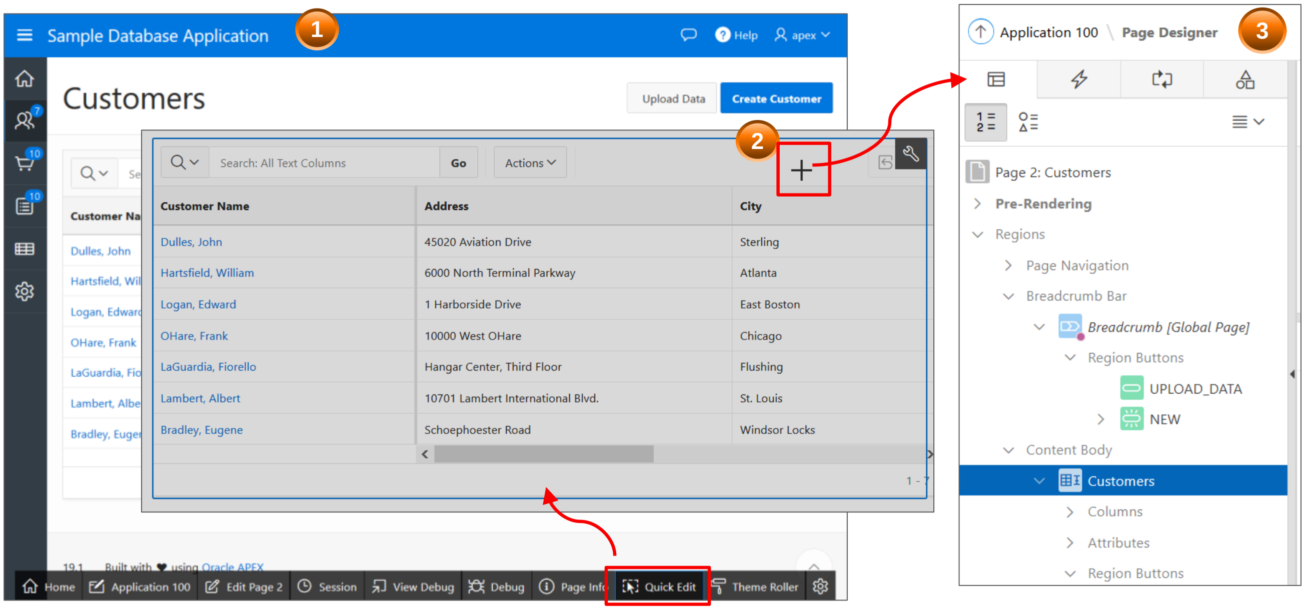
Page Items to Submit

Now that you have created/added new regions to your page, let's learn how to edit the region.

To edit a region, select the region you want to edit in the Rendering pane of the Page Designer. Alternatively, you can select the region from the Layout. As soon as you select the region, the Property Editor loads the region attributes in the right pane. Use the Property Editor to edit the attributes and save and run the page.

In the next couple of slides, you learn how to edit a region at run time, by using the Quick Edit and Live Template Options (you learned about these Runtime Developer Toolbar options a few slides back).

Editing a Region at Runtime (using Quick Edit option)



Customer Name	Address	City
Dulles, John	45020 Aviation Drive	Sterling
Hartsfield, William	6000 North Terminal Parkway	Atlanta
Logan, Edward	1 Harborside Drive	East Boston
OHare, Frank	10000 West OHare	Chicago
LaGuardia, Fiorello	Hangar Center, Third Floor	Flushing
Lambert, Albert	10701 Lambert International Blvd.	St. Louis
Bradley, Eugene	Schoephoester Road	Windsor Locks

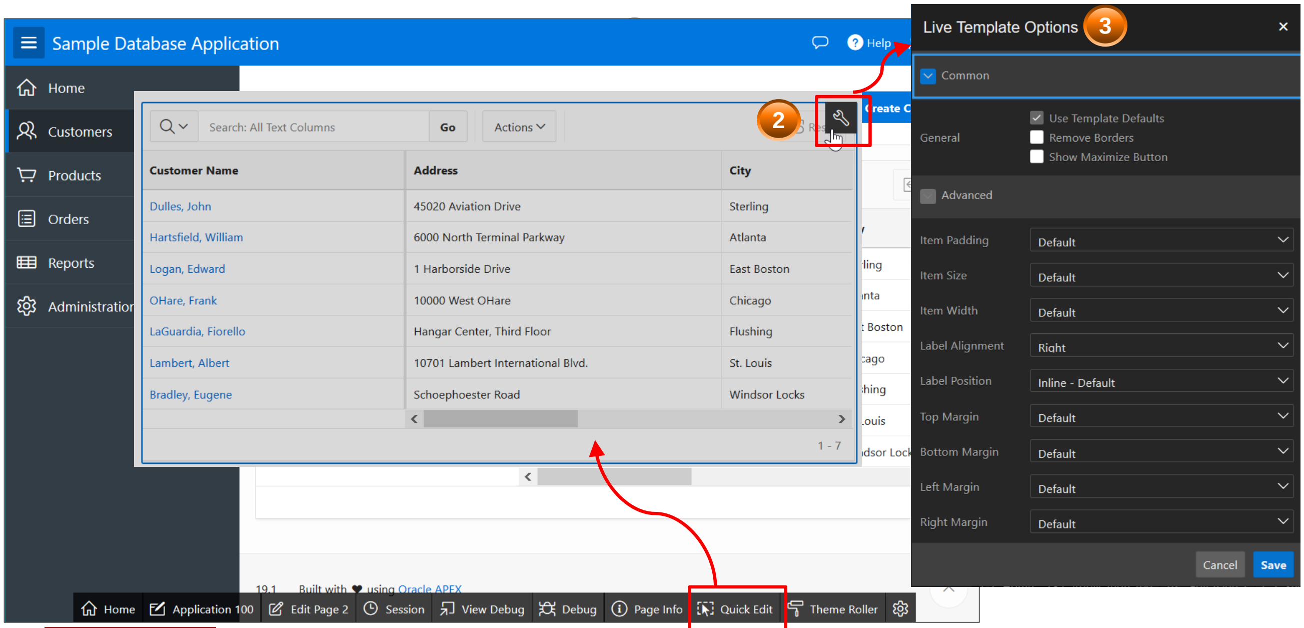
To edit a region at Runtime:

1. Run your application.
2. Click **Quick Edit** on the Developer Toolbar.
3. Select the desired page component.
4. Click anywhere on the page component, and it takes you instantly to the Page Designer view of the selected component. You can now make the necessary changes. In the screenshot, Page 2 – Customers page is selected.

   Note that to exit quick edit mode, you must press ESCAPE or click outside a component.

   In the next slide, you learn how to edit a region at Runtime using the Live Template Options (via Quick Edit on the Developer Toolbar).

Editing a Region at Runtime (using Live Template Options)



Customer Name	Address	City
Dulles, John	45020 Aviation Drive	Sterling
Hartsfield, William	6000 North Terminal Parkway	Atlanta
Logan, Edward	1 Harborside Drive	East Boston
OHare, Frank	10000 West OHare	Chicago
LaGuardia, Fiorello	Hangar Center, Third Floor	Flushing
Lambert, Albert	10701 Lambert International Blvd.	St. Louis
Bradley, Eugene	Schoephoester Road	Windsor Locks

Live Template Options ×

☑ Common

General
☑ Use Template Defaults
□ Remove Borders
□ Show Maximize Button

□ Advanced

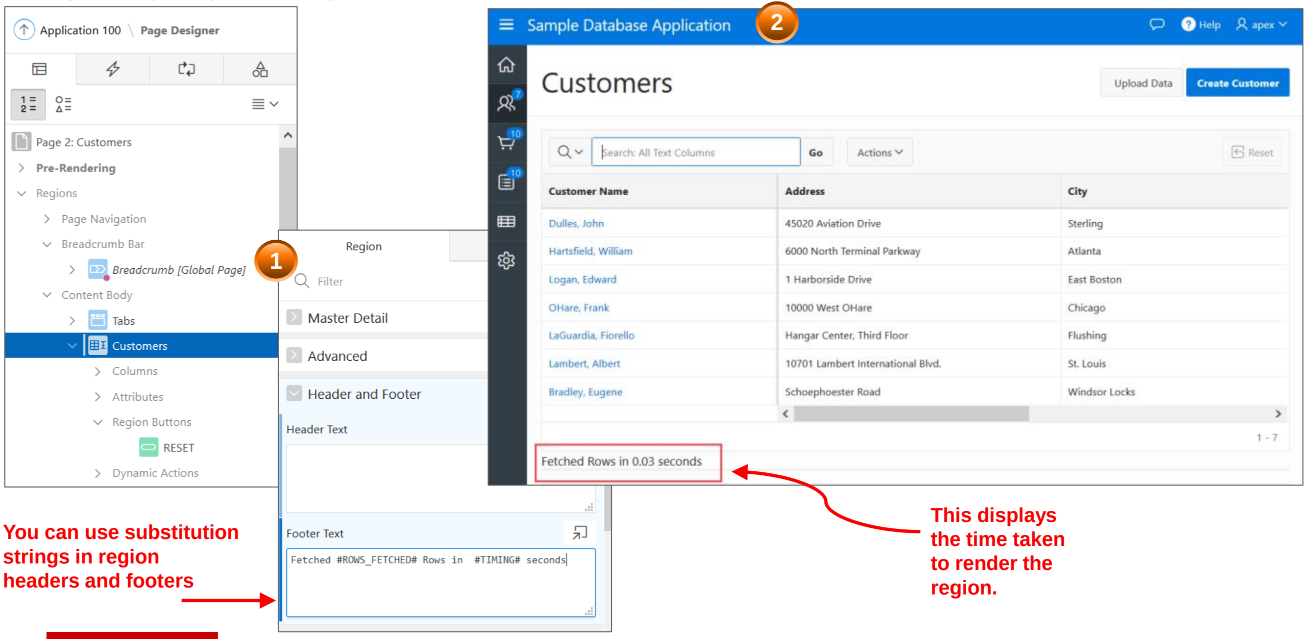
Item Padding	Default
Item Size	Default
Item Width	Default
Label Alignment	Right
Label Position	Inline - Default
Top Margin	Default
Bottom Margin	Default
Left Margin	Default
Right Margin	Default

Cancel Save

To edit a region at Runtime using Live Template Options:

1. Run your application.
2. Click **Quick Edit**.
3. Move the mouse over the component for which you want to modify template options. In the screenshot, the Customers page is selected.
4. Click the Wrench icon in the upper-right corner.
5. The **Live Template Options** window appears. You can make the necessary changes here and click **Save**.

Specifying a Region Header and Footer



```
Fetched #ROWS_FETCHED# Rows in  #TIMING# seconds
```

Customer Name	Address	City
Dulles, John	45020 Aviation Drive	Sterling
Hartsfield, William	6000 North Terminal Parkway	Atlanta
Logan, Edward	1 Harborside Drive	East Boston
OHare, Frank	10000 West OHare	Chicago
LaGuardia, Fiorello	Hangar Center, Third Floor	Flushing
Lambert, Albert	10701 Lambert International Blvd.	St. Louis
Bradley, Eugene	Schoephoester Road	Windsor Locks

Fetched Rows in 0.03 seconds

You can use substitution strings in region headers and footers

This displays the time taken to render the region.

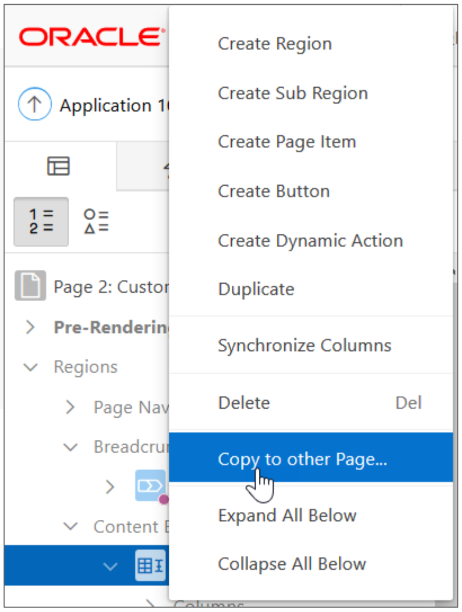
Now, let’s try working a bit more on the pages and regions – maybe specify a region header and footer. Substitution strings can be used in region headers and footers.

In this slide, Steve uses `#ROWS_FETCHED#` and `#TIMING#` substitution strings (*Fetched #ROWS_FETCHED# Rows in #TIMING# seconds.*) in the region (*Customers*) footer. These substitution strings calculate the number of rows fetched in the elapsed time in seconds when rendering a region.

Copying a Region

1. In the Rendering pane, right-click the region node and select **Copy to other Page**.
2. Specify the page where you want to copy the region.
3. You also have an option to copy the buttons.
4. Click **Next >**.
5. Enter the new region name and click **Copy**.

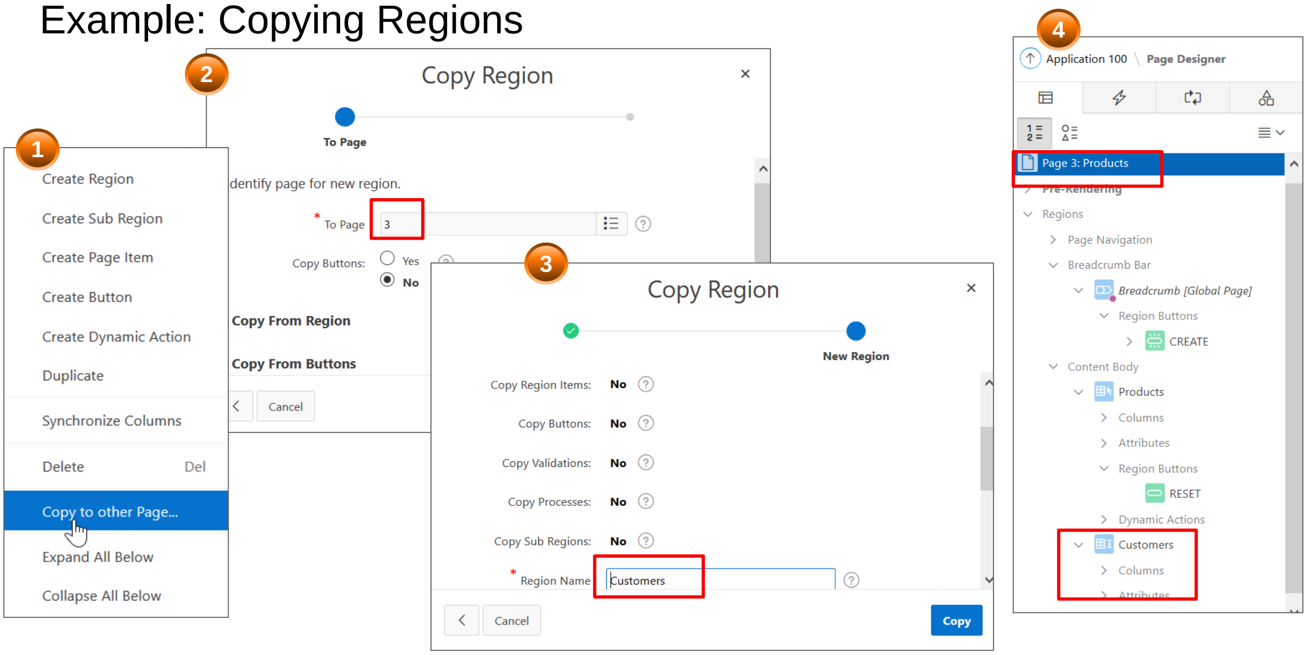
   The region is copied to the specified page.



This slide provides you an overview of the steps to copy a region. You can copy regions from one page to another within an application. When copying, you can include the region items and buttons as well.

Let’s see how in the next slide.

Example: Copying Regions



Identify page for new region.

* To Page: 3

Copy Buttons: ○ Yes ● No

Copy Region Items: No

Copy Buttons: No

Copy Validations: No

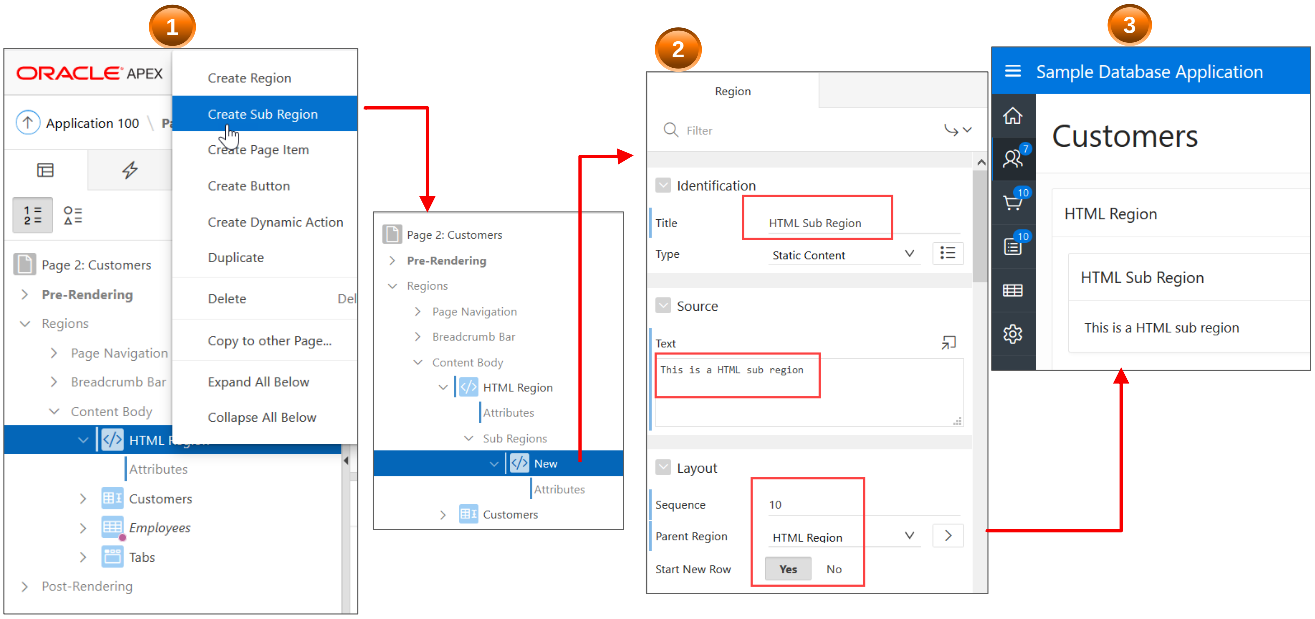
Copy Processes: No

Copy Sub Regions: No

* Region Name: Customers

In the example in the slide, Steve is copying the *Customers* page region. He wants the same region attributes again and does not want to repeat the task of creating a new region. So, he uses this feature to replicate the region he created a little while back.

Creating a Subregion



Title: HTML Sub Region
Type: Static Content
Text: This is a HTML sub region
Sequence: 10
Parent Region: HTML Region
Start New Row: [✓ Yes] No

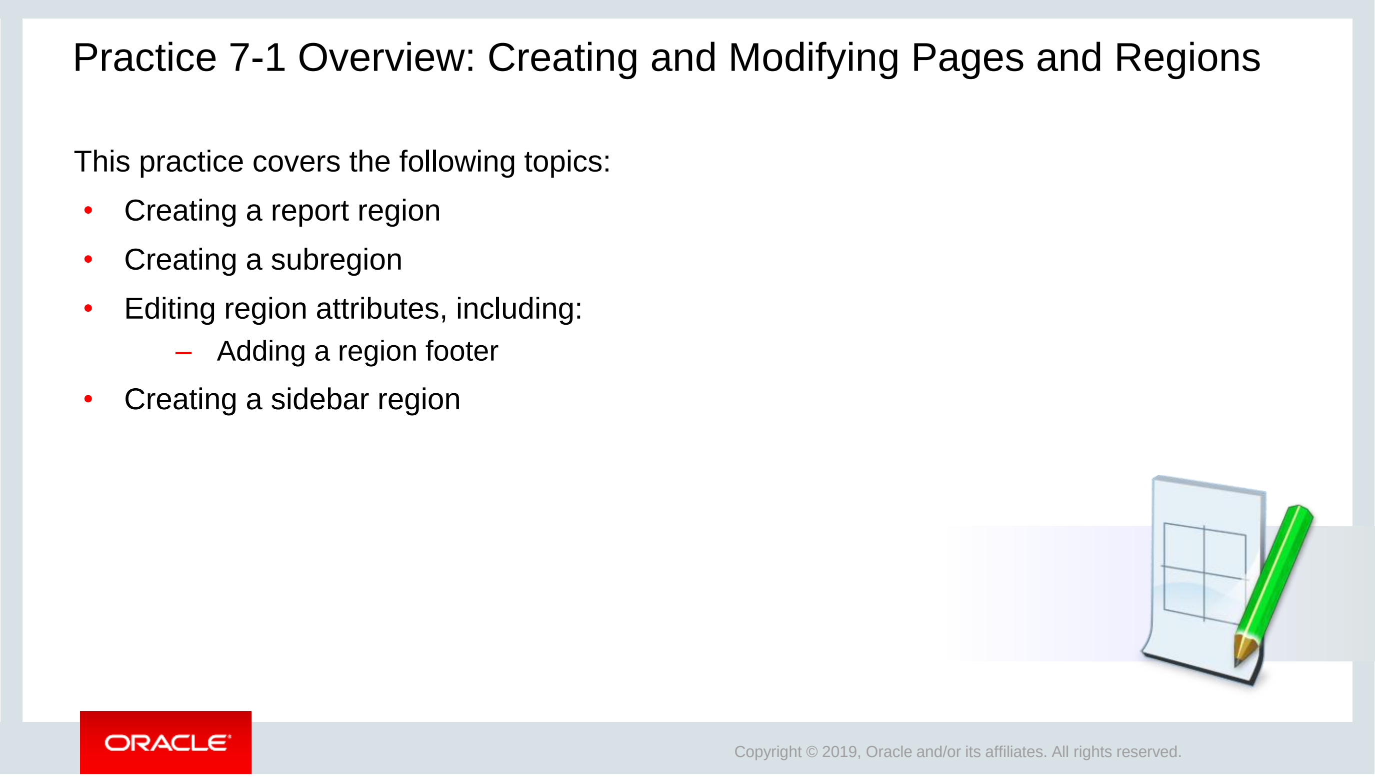
Steve creates a region called **HTML Region** on the *Customers* page, and he wants to add another region inside that – basically, wants to create a subregion.

A subregion is a region within another region. You must specify a parent region when you are creating this region. In the slide above, Steve has selected HTML Region as his parent region under Layout.

Perform the following steps:

1. In the Rendering pane, right-click the node for the region where you want a subregion and select **Create Sub Region**.
2. The subregion is created. Specify the region attributes in the Property Editor. Note that the Parent Region field is automatically set to the region node that you selected in the previous step.
3. Save and run the page.

Note: A subregion can also be created by dragging a region from the Regions gallery to inside the parent region.



Practice 7-1 Overview: Creating and Modifying Pages and Regions

This practice covers the following topics:

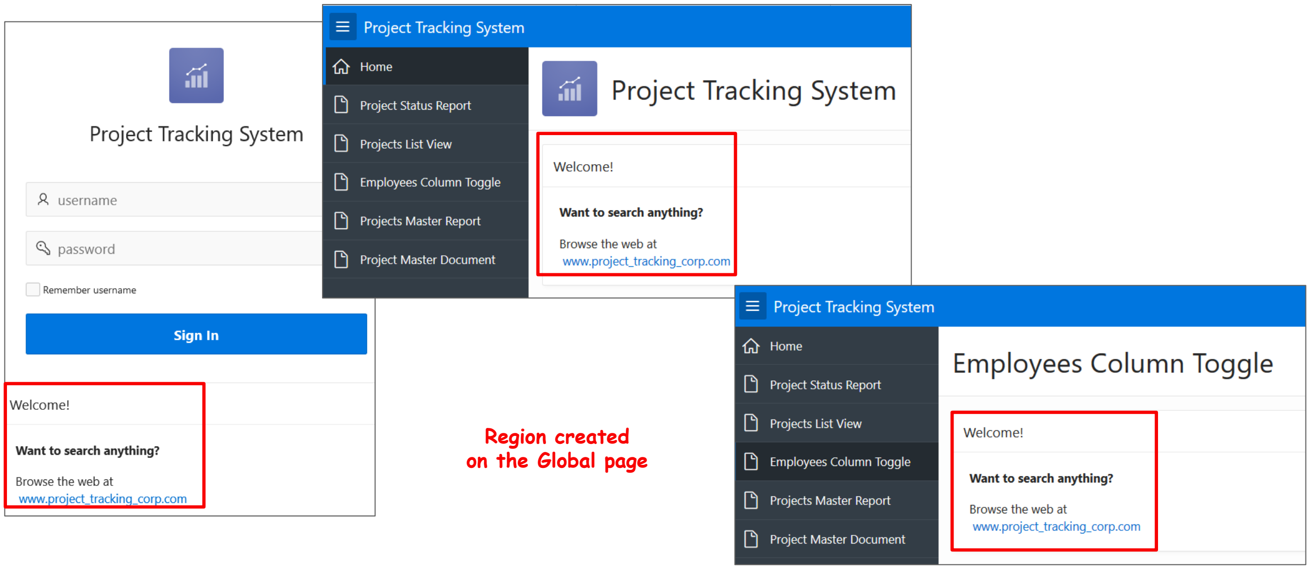
- Creating a report region
- Creating a subregion
- Editing region attributes, including:
  - Adding a region footer
- Creating a sidebar region

Lesson Agenda

- Introducing Page Definition
- Working with Page Regions
- Working with Pages
  - What Is a Global Page?
  - Creating a Global Region
  - Viewing Pages of the Application
  - Creating and Using Page Groups
  - Copying a Database Application Page
  - Deleting a Page

Global Page

Any item, button, or region on this page is displayed on all the pages in the application.



Welcome!

Want to search anything?

Browse the web at www.project_tracking_corp.com

Region created on the Global page

Both *Project Status Report* and *Employees Column Toggle* report open with the same *Welcome* message and link to the organization website that Steve and Stella are working in.

It was done by modifying the Global page of the application.

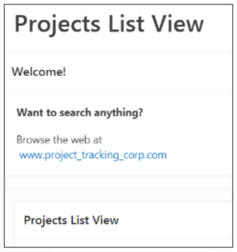
The Global page of your application functions as a master page and gets created automatically as and when you create your application. The Application Express engine renders all components you add to a Global page on every page within your application.

Note: You cannot create processes, computations, or branches on the Global page. You will learn about all these features in later lessons.

Creating a Global Page Region

1. Navigate to the **Global** page of the application.
2. Double-click to edit the page in Page Designer. The definition of the Global page opens in Page Designer.

   The page definition for a Global page looks different from other pages.
3. Under Regions, right-click and select Create Region. Enter the Title, Type, and Source Text to Display. Choose the display position from the Layout.
4. Save the page. Note that you cannot run a **Global** page directly.
5. Run the application to see the newly added text.



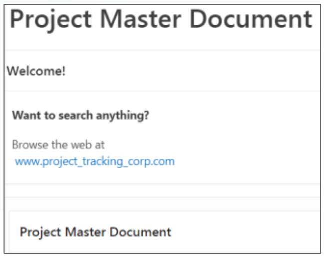
Projects List View

Welcome!

Want to search anything?

Browse the web at
www.project_tracking_corp.com

Projects List View



Project Master Document

Welcome!

Want to search anything?

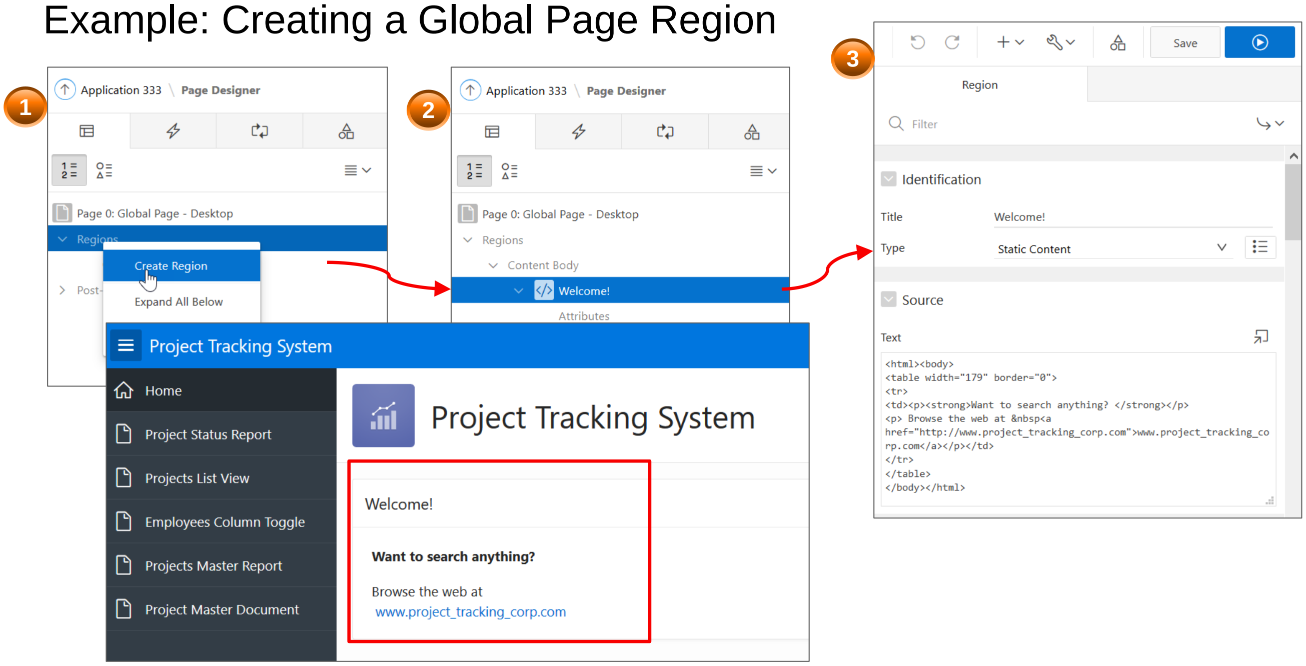
Browse the web at
www.project_tracking_corp.com

Project Master Document

This slide provides an overview of the steps to create a Global Region.

Steve is very interested in this particular feature of Oracle Application Express. He wants the link to his company website appearing on all his application pages. Therefore, the *Global* page is an obvious choice. He now switches from the *Sample Database Application* and starts working on his *PTS* application.

Example: Creating a Global Page Region



Identification

Title: Welcome!

Type: Static Content

Source

Text

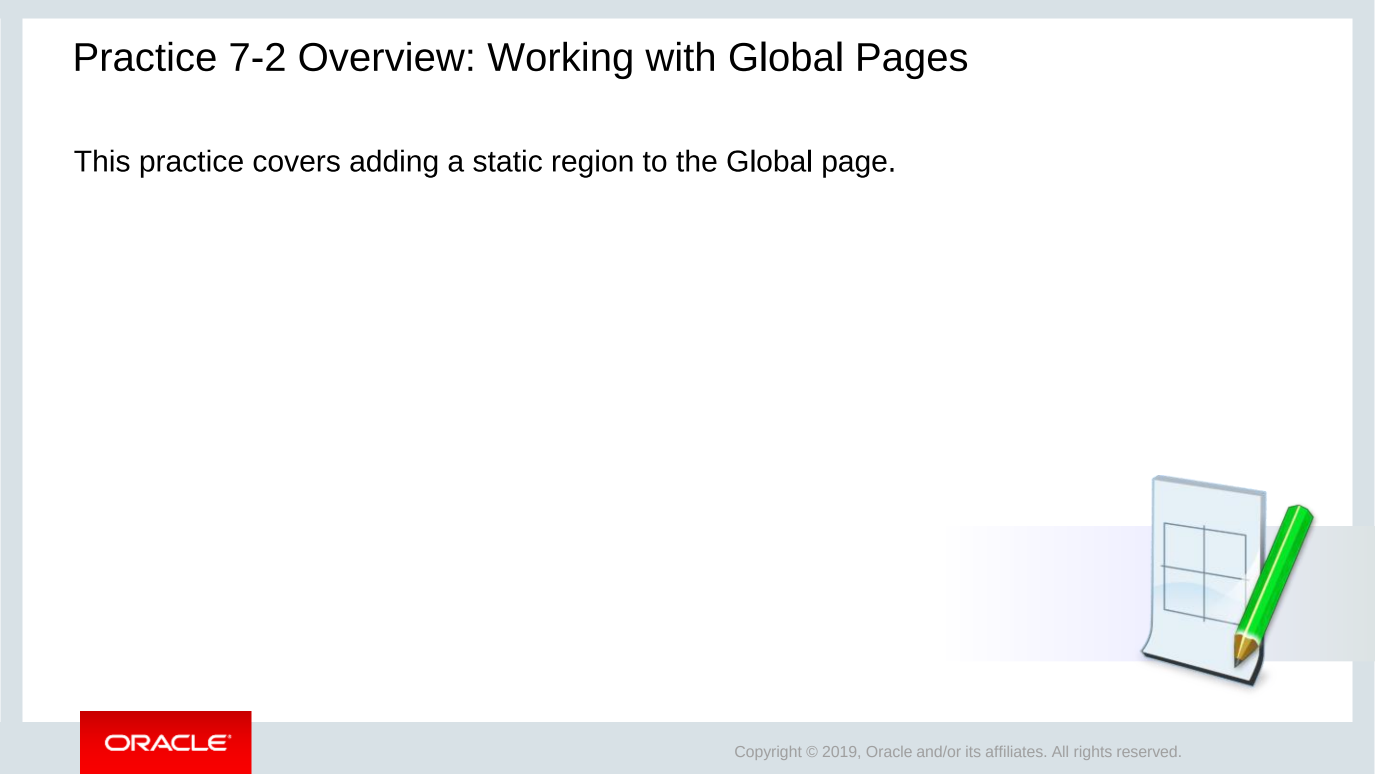
```
<html><body>
<table width="179" border="0">
<tr>
<td><p><strong>Want to search anything? </strong></p>
<p> Browse the web at &nbsp<a
href="http://www.project_tracking_corp.com">www.project_tracking_co
rp.com</a></p></td>
</tr>
</table>
</body></html>
```

This slide shows an example of how to create a region on the *Global* page. When you create an application, apart from the *Home* page and the *Login* page, a *Global* page is automatically created. As you see in this slide, Steve first creates a Static region type on the Global page. He then adds the text he wants to display in the *Source* field. By defining the region on the Global page, the region is displayed on all pages in his application. Note that you can also restrict the region to appear only on certain pages. Let's see how Steve created the Global region:

1. Open **Global p**age in Page Designer view.
2. Under **Rendering**, right-click **Regions** and select *Create Region*.
3. In the Properties Pane on the right, enter the following values:
   - **Identification**: **Title**: `Welcome!`
   - **Identification**: **Type**: `Static Content`
   - **Source** (code text area):

```
<html><body>
<table width="179" border="0">
<tr>
<td><p><strong>Want to search anything? </strong></p>
<p> Browse the web at &nbsp<a
href="http://www.project_tracking_corp.com">www.project_tracking_corp.
com</a></p>\\\\\\\\ </td>
</tr>
</table>
</body></html>
```

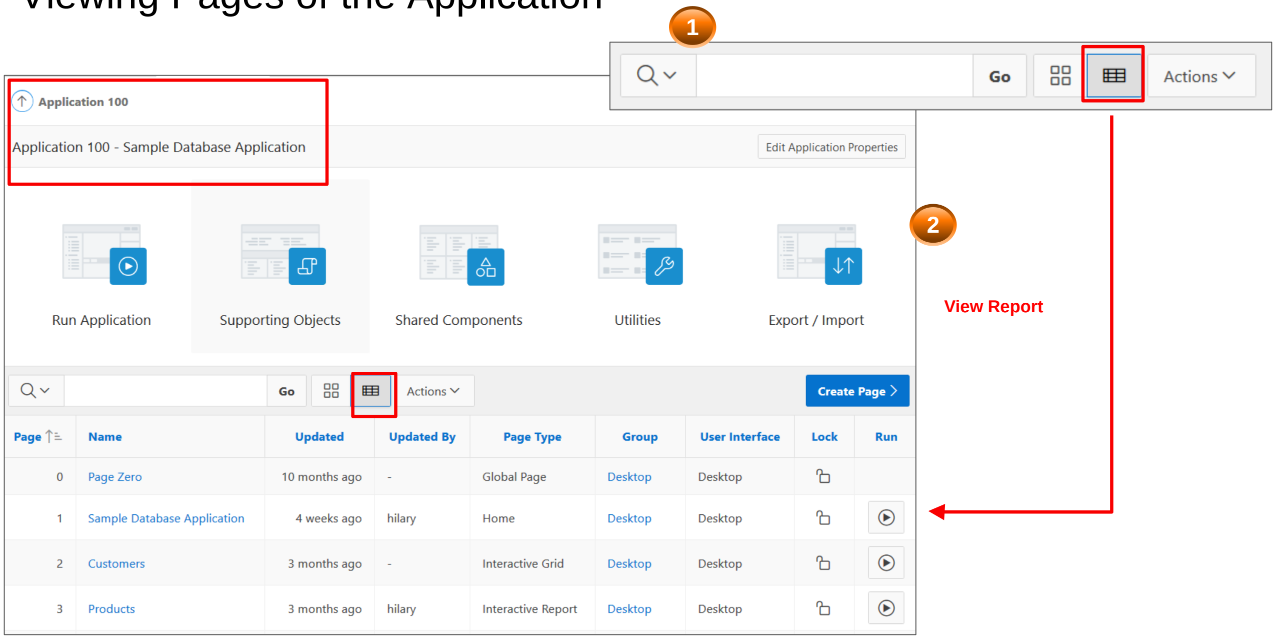
4. Click **Save**. Run your application. You can see a box with content displayed on all pages of the application including the **Sign in** Page. Note that Global page cannot be run directly.



Practice 7-2 Overview: Working with Global Pages

This practice covers adding a static region to the Global page.

Viewing Pages of the Application



Application 100 - Sample Database Application

Page	Name	Updated	Updated By	Page Type	Group	User Interface	Lock	Run
0	Page Zero	10 months ago	-	Global Page	Desktop	Desktop		
1	Sample Database Application	4 weeks ago	hilary	Home	Desktop	Desktop		
2	Customers	3 months ago	-	Interactive Grid	Desktop	Desktop		
3	Products	3 months ago	hilary	Interactive Report	Desktop	Desktop		

To view all the pages in your application, on the Application home page, change the view to **View Report**. The slide above lists all the pages belonging to the Sample Database Application.

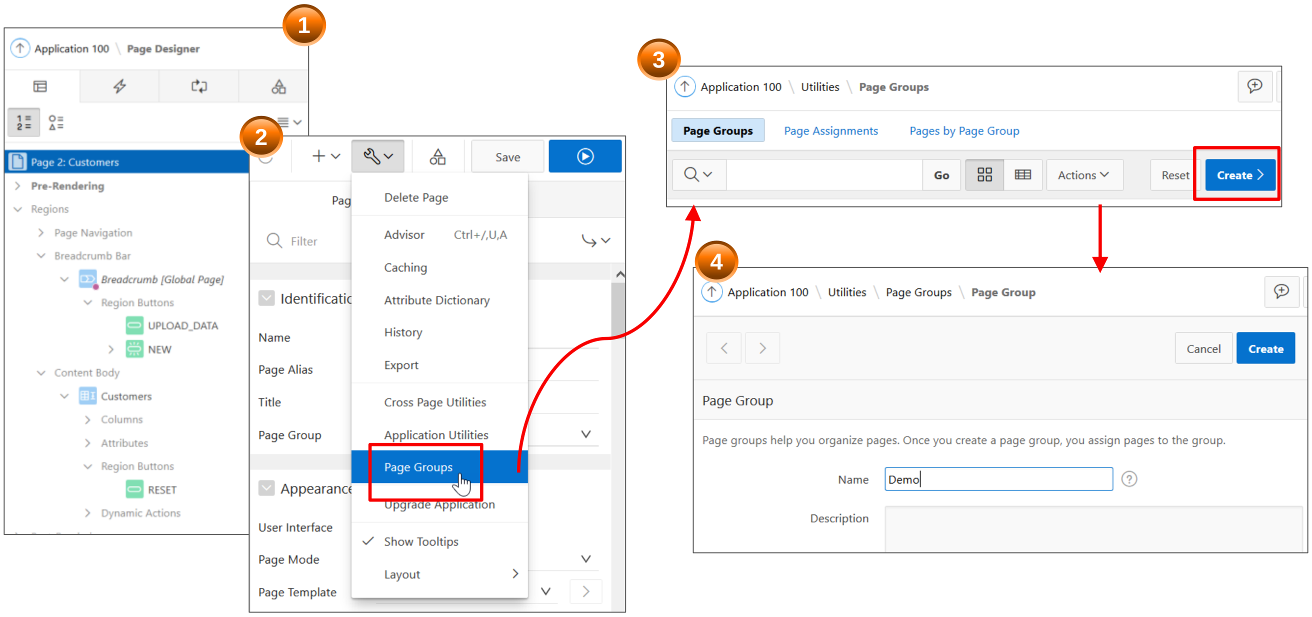
You can also sort the pages according to the page groups they are assigned to. You will learn about page groups and how to create a page group in the following slide.

Creating a Page Group

1. From the page definition of any page, click the down arrow on the Utilities button and select Page Groups.
2. Click the Create button.
3. Enter the name of the page group and click Create. The Page Group is created.
4. After you have created page groups, assign pages to them.

This slide provides an overview of the steps to create a *Page Group*. Page groups help you to organize pages. To use page groups, you must create a group and then assign pages to this group. This will help you list all the corresponding pages belonging to a particular group together. The next slide shows you how to create a page group.

Example: Creating a Page Group



Page groups help you organize pages. Once you create a page group, you assign pages to the group.

Name: Demo

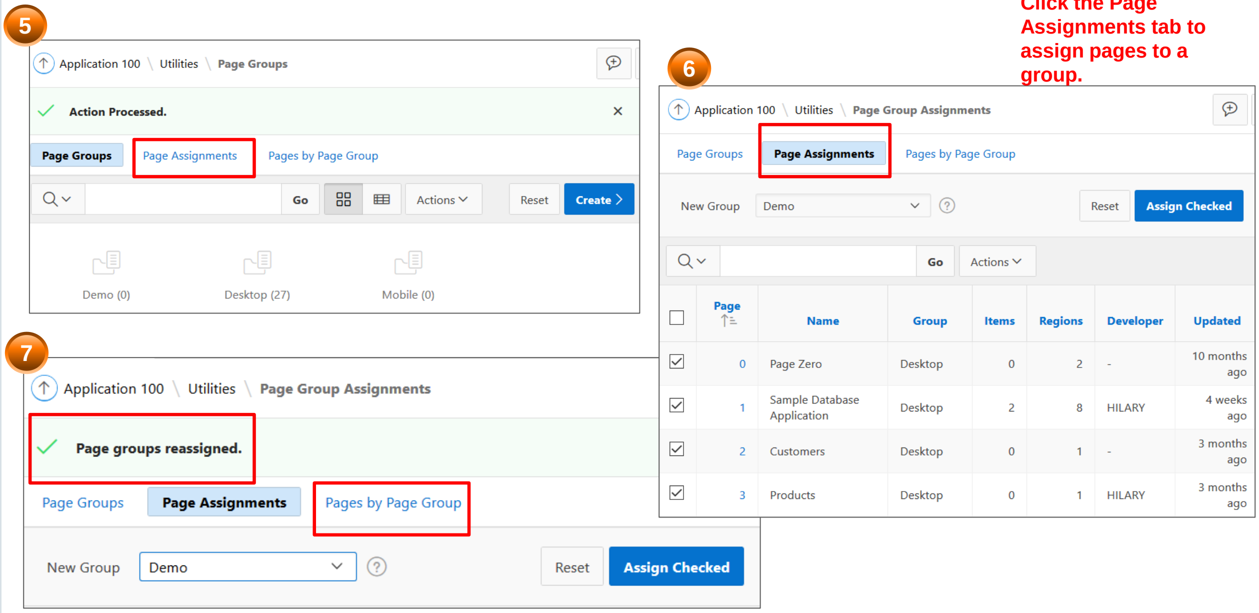
Description:

In the previous slide, you had an overview of how to create a page group. Let’s try assigning the page group now.

To assign pages to a group:

1. Go to the Utility menu and select **Page Groups** (screenshot 2).
2. In Page Groups, click **Create** (screenshot 3).
3. Enter a name for the new page group in the **Name** field and click **Create** (screenshot 4).

Example: Creating a Page Group



Action Processed.

☐	Page	Name	Group	Items	Regions	Developer	Updated
☑	0	Page Zero	Desktop	0	2	-	10 months ago
☑	1	Sample Database Application	Desktop	2	8	HILARY	4 weeks ago
☑	2	Customers	Desktop	0	1	-	3 months ago
☑	3	Products	Desktop	0	1	HILARY	3 months ago

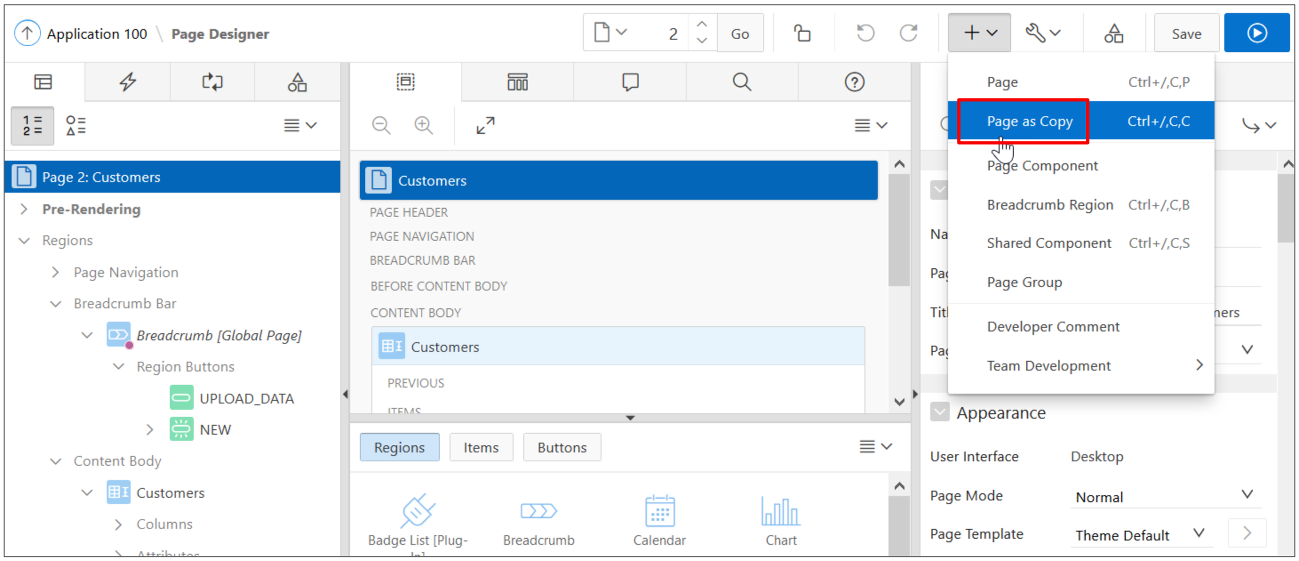
Page groups reassigned.

Click the Page Assignments tab to assign pages to a group.

4. After the new page group is created, the message “Action Processed” is displayed (screenshot 5). Click the **Page Assignments** tab and select the group from the New Group-Demo list (screenshot 6).
5. Select the check box next to the pages that you want to assign to the group and click the **Assign Checked** button (screenshot 6).

After you have created page groups and assigned pages to them, you can view the page group by clicking **Pages by Page Group** (screenshot 7).

Copying a Database Application Page



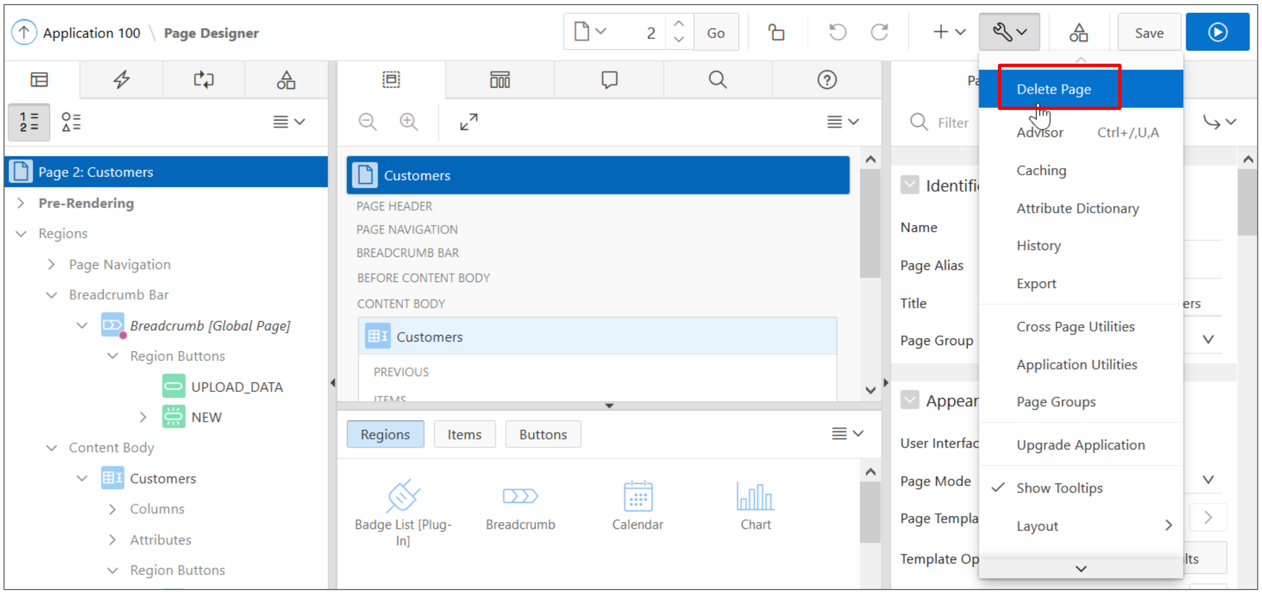
To save time and effort, Steve wants to try using the Copy Page feature of Oracle Application Express. Let us see how he can copy an existing page.

You can copy a page either from the current application or from another application. During the copy process, you can also copy shared components or change mappings to shared components in the target application.

To copy a page:

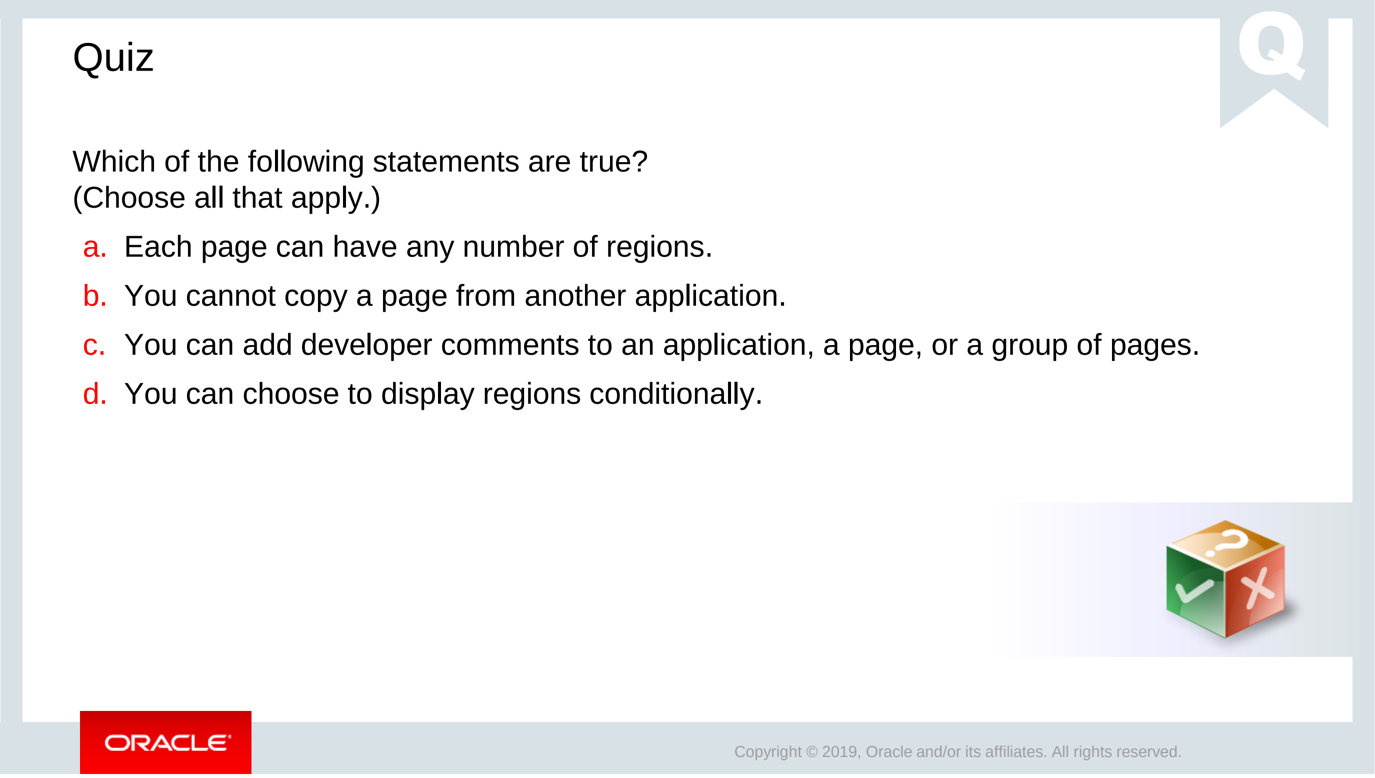
1. Open the Page Definition of the page you want to copy.
2. On the Page Designer toolbar, click the **Create** menu and then select **Page as Copy**.
3. On Copy Page Option, select one of the following:
   - **Page in this application**
   - **Page in another application**
4. Follow the on-screen instructions.

Deleting a Page



In this slide, you learn how to delete a page from your application. By deleting a page, you also delete the corresponding tabs, breadcrumbs, and list entries. To delete a page in Page Designer:

1. View the page in Page Designer.
2. Verify the page number.
3. On the Page Designer toolbar, click **Utilities** and select **Delete Page**.
4. Click **Permanently Delete Page**.



Quiz

Which of the following statements are true?
(Choose all that apply.)

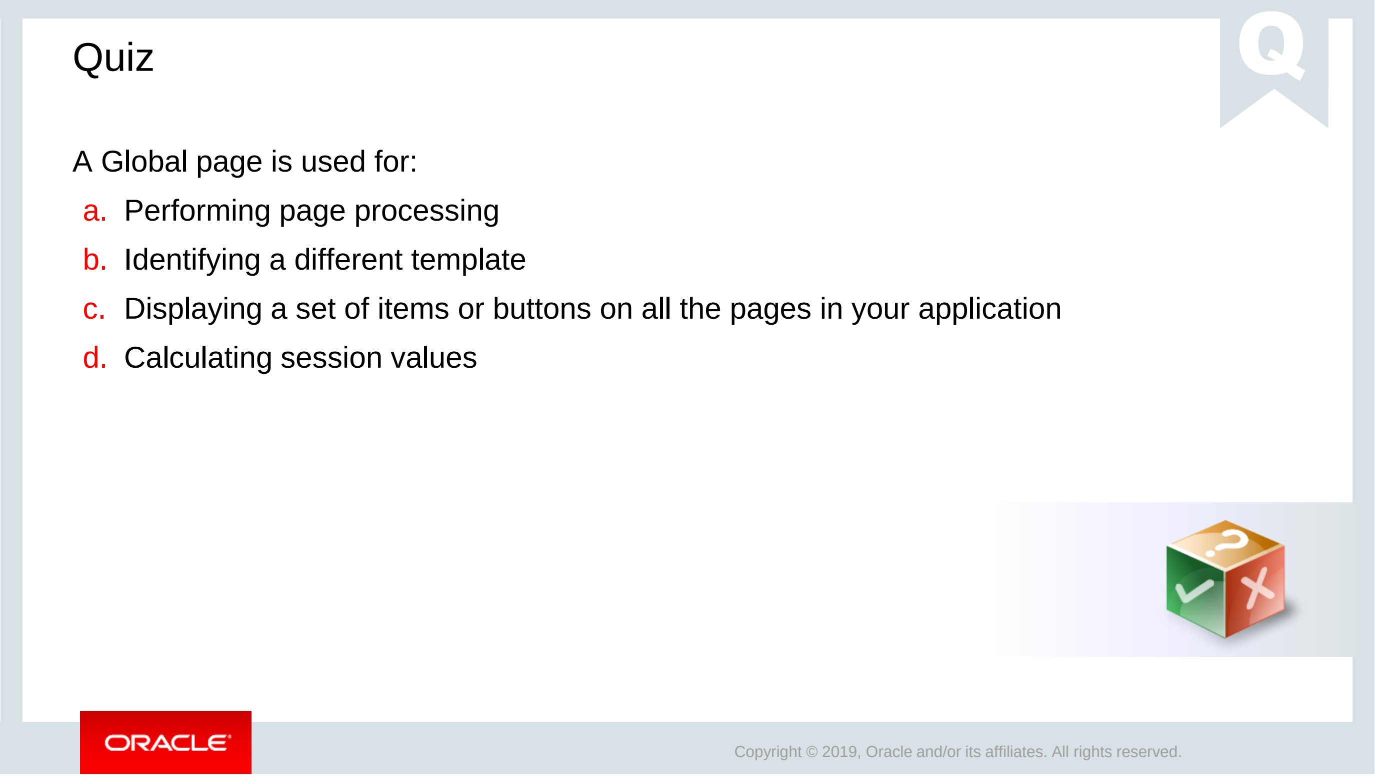
a. Each page can have any number of regions.

b. You cannot copy a page from another application.

c. You can add developer comments to an application, a page, or a group of pages.

d. You can choose to display regions conditionally.

Answer: a, c, d



Quiz

A Global page is used for:

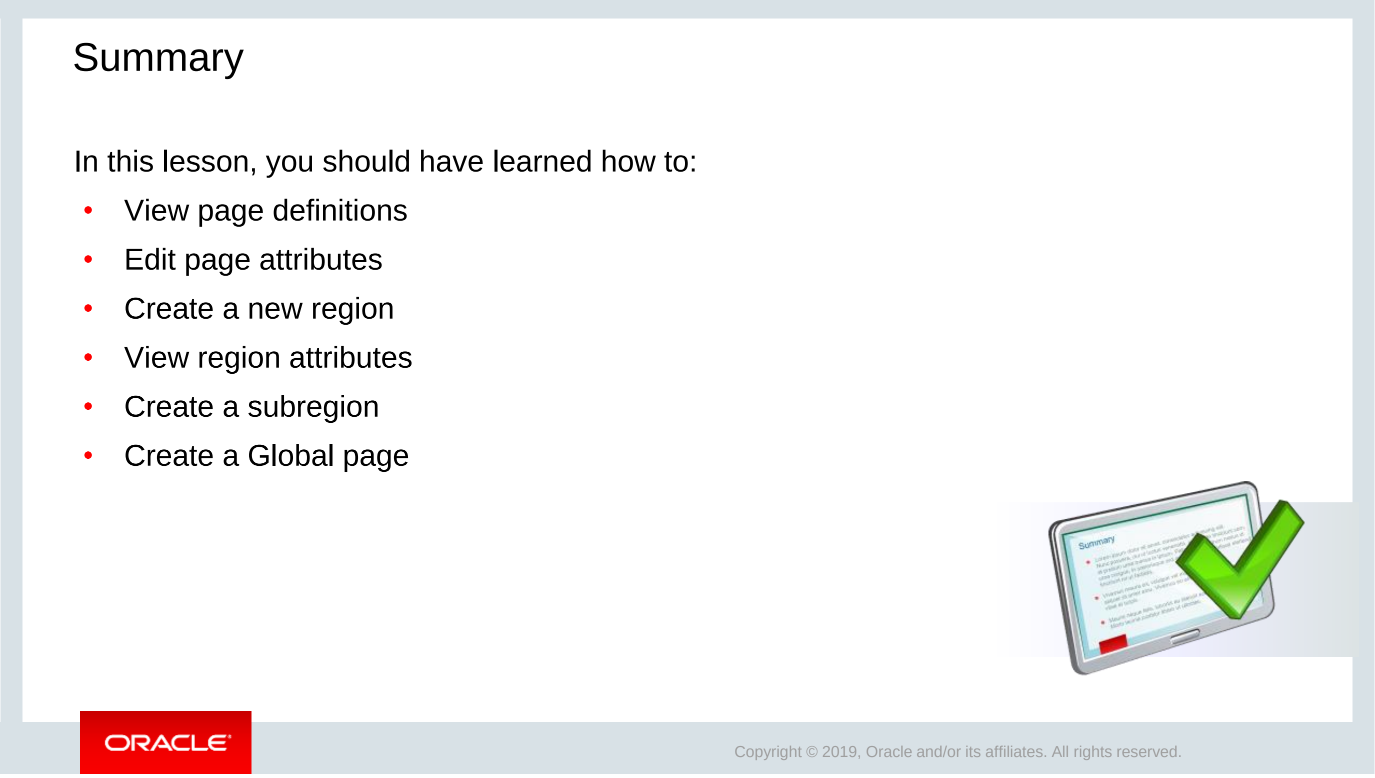
a. Performing page processing

b. Identifying a different template

c. Displaying a set of items or buttons on all the pages in your application

d. Calculating session values

Answer: c



Summary

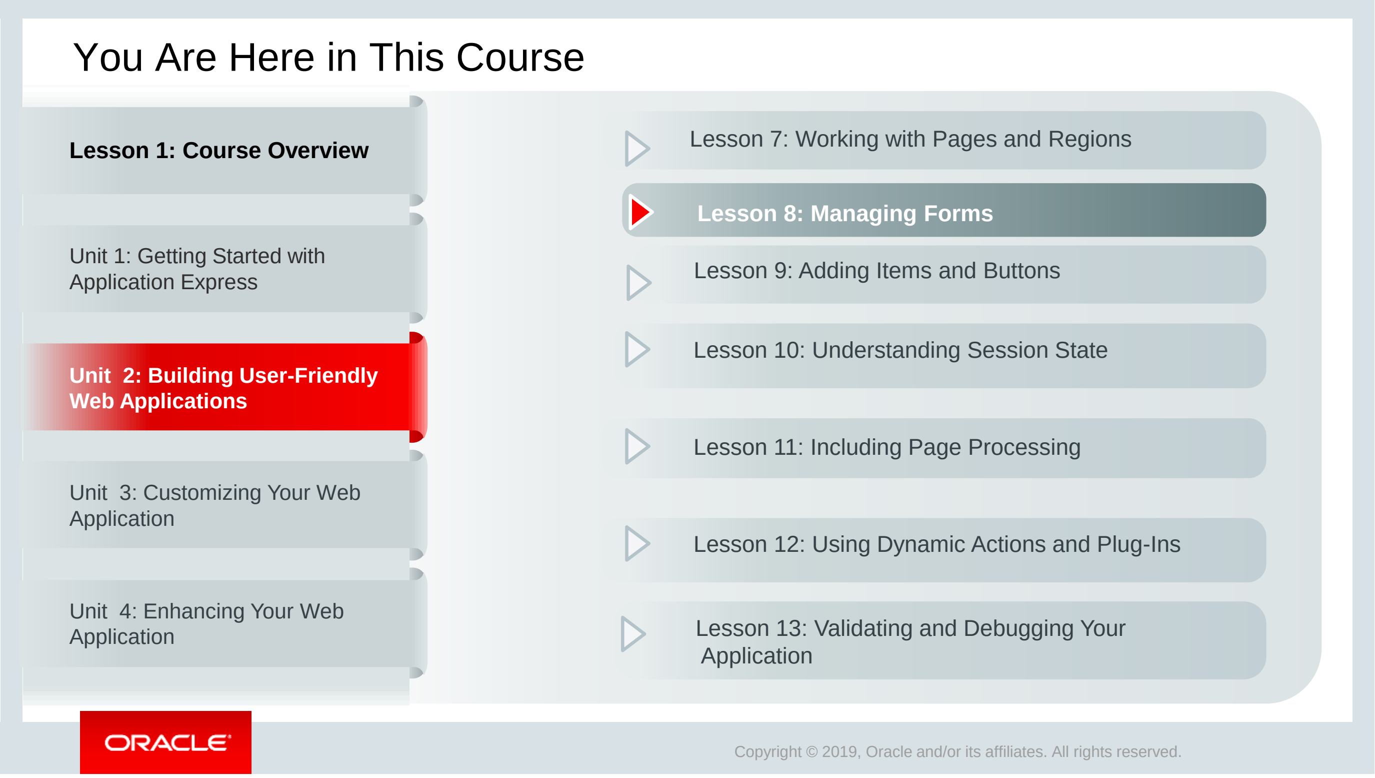
In this lesson, you should have learned how to:

- View page definitions
- Edit page attributes
- Create a new region
- View region attributes
- Create a subregion
- Create a Global page

In this lesson, you learned about pages and regions. You learned how to create pages and regions and how to edit their attributes.

8

Managing Forms



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

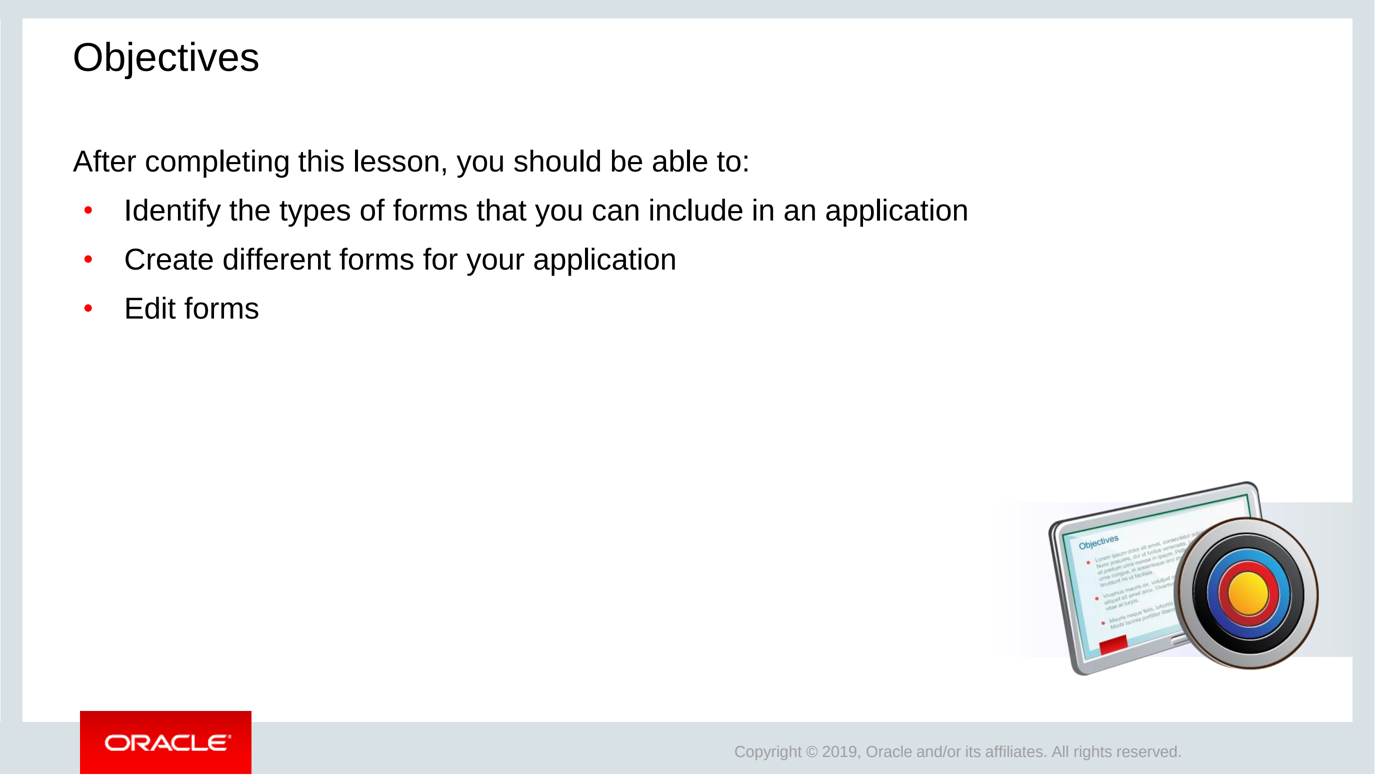
Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- Lesson 7: Working with Pages and Regions
- **Lesson 8: Managing Forms**
- Lesson 9: Adding Items and Buttons
- Lesson 10: Understanding Session State
- Lesson 11: Including Page Processing
- Lesson 12: Using Dynamic Actions and Plug-Ins
- Lesson 13: Validating and Debugging Your Application

ORACLE

This slide is a graphical depiction of the course, particularly highlighting Unit 2 - Lesson 8, which is dealt with in these slides.

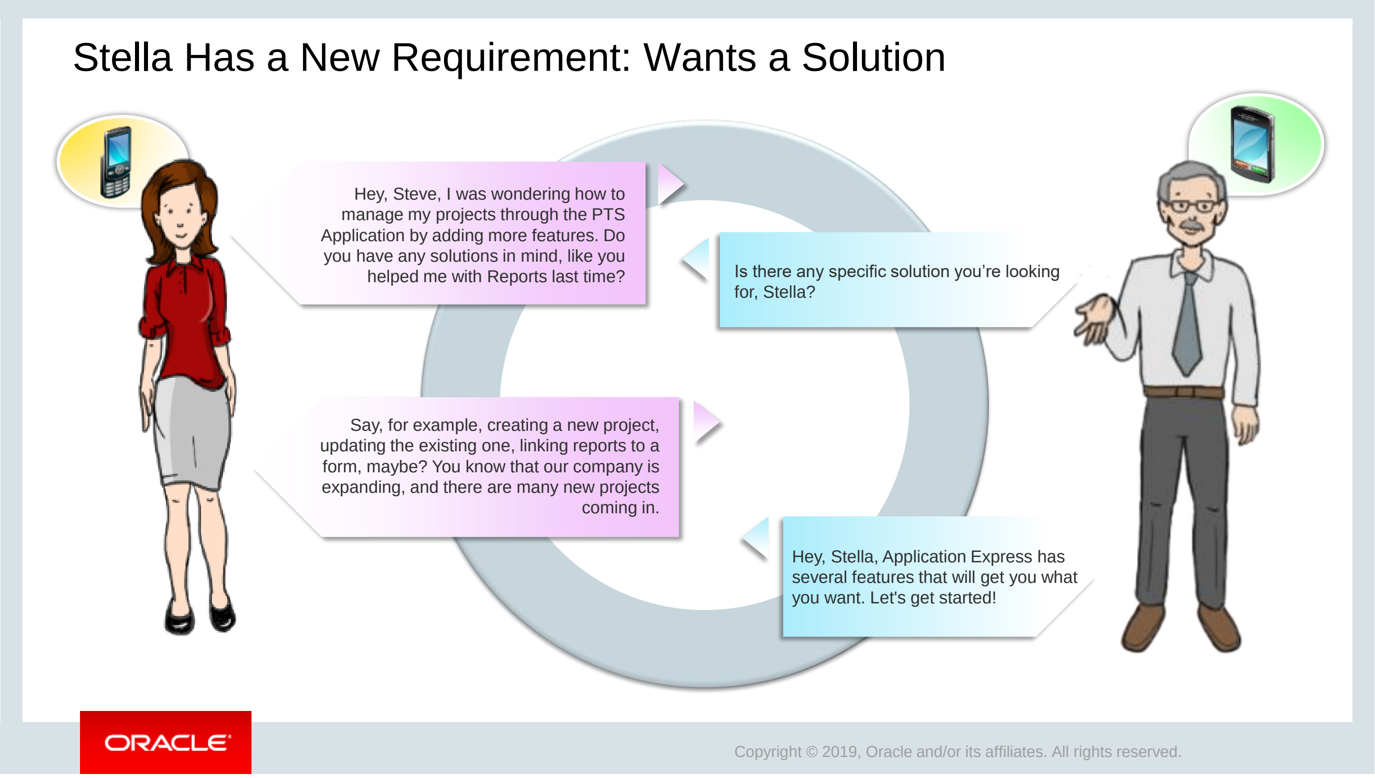


Objectives

After completing this lesson, you should be able to:

- Identify the types of forms that you can include in an application
- Create different forms for your application
- Edit forms

In this lesson, you learn how to create forms in your application by using the various built-in wizards. You also learn how to edit and modify forms.



Stella Has a New Requirement: Wants a Solution

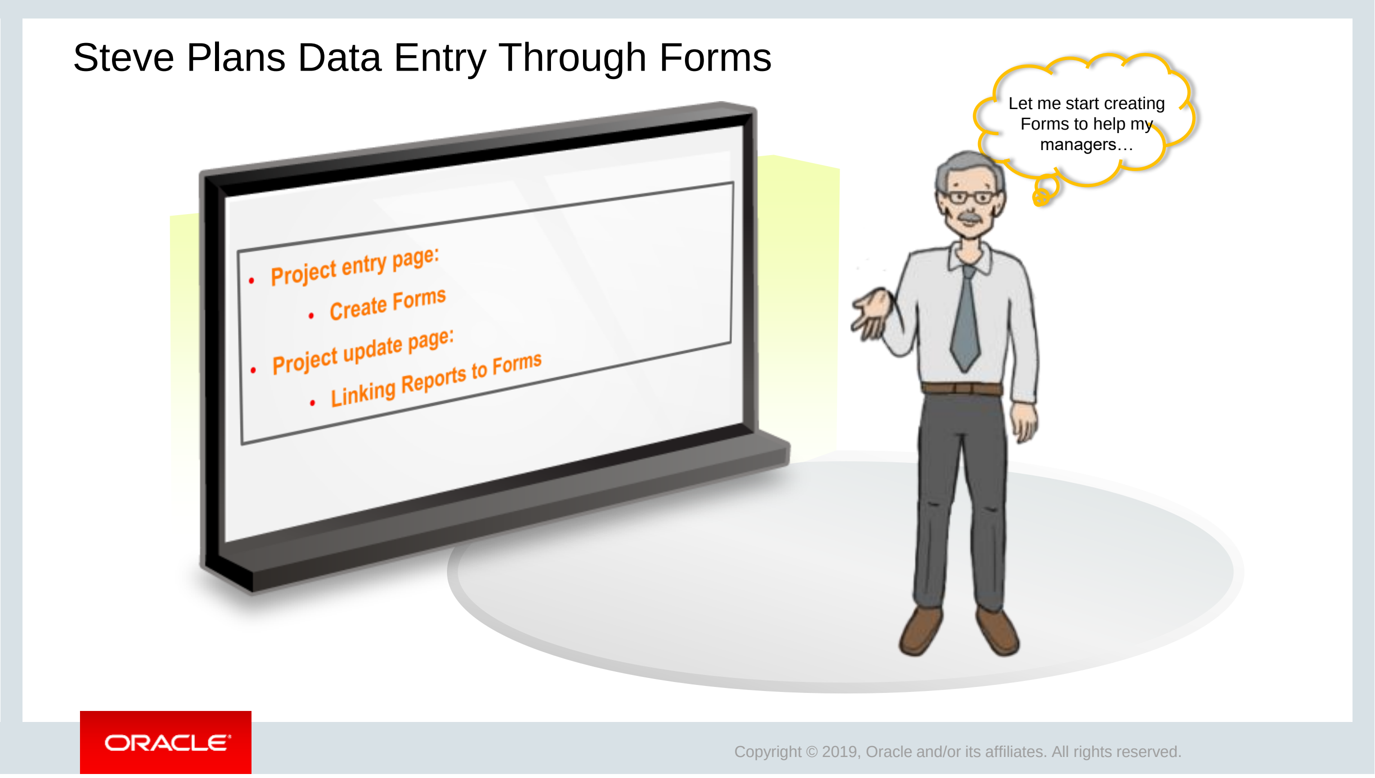
Hey, Steve, I was wondering how to manage my projects through the PTS Application by adding more features. Do you have any solutions in mind, like you helped me with Reports last time?

Is there any specific solution you’re looking for, Stella?

Say, for example, creating a new project, updating the existing one, linking reports to a form, maybe? You know that our company is expanding, and there are many new projects coming in.

Hey, Stella, Application Express has several features that will get you what you want. Let's get started!

Stella expresses her concern about the need for creating new projects, updating existing projects, and also linking her Reports to Forms. Steve assures that Oracle Application Express has exactly what she is looking for.

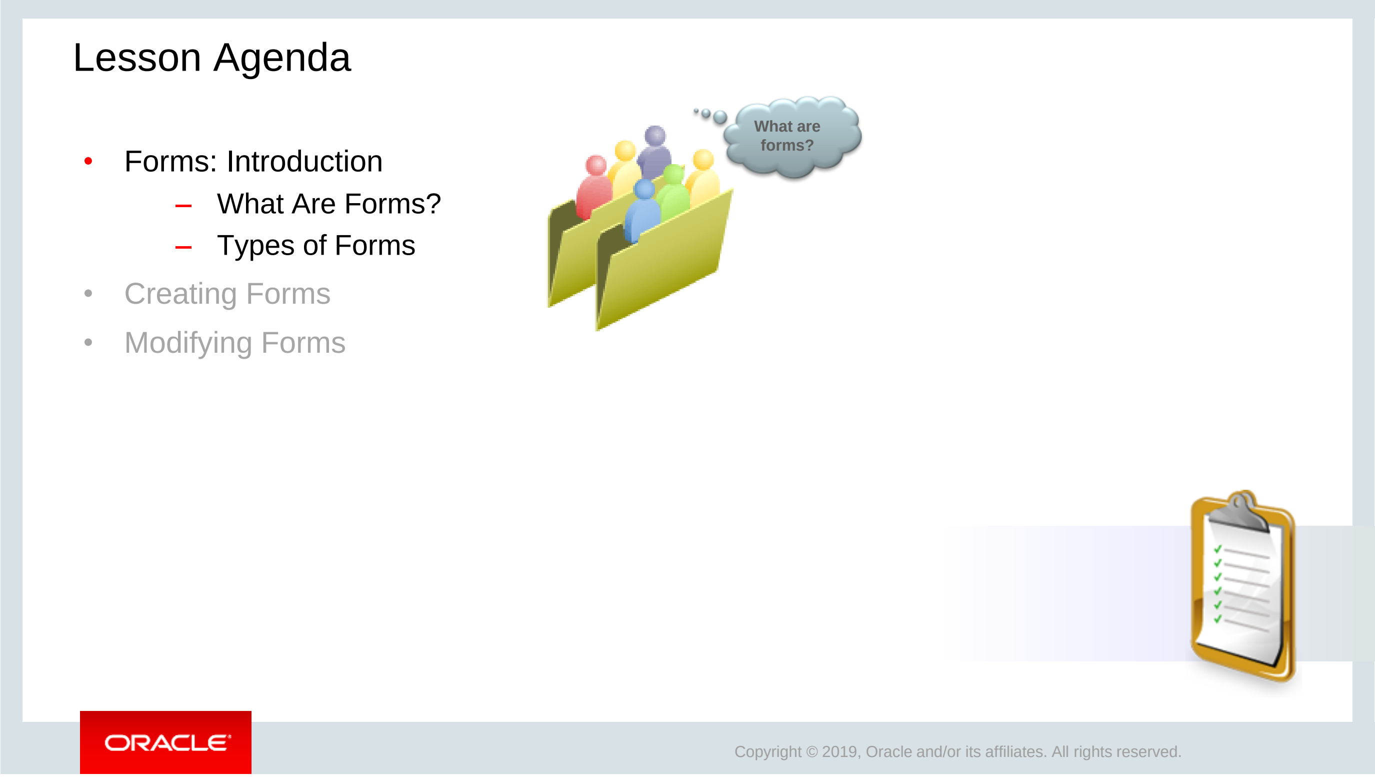


Steve Plans Data Entry Through Forms

- Project entry page:
- Create Forms
- Project update page:
- Linking Reports to Forms
Let me start creating Forms to help my managers…
ORACLE

Stella wants the employees database to be updated with all new employee details so that she can allocate employees to various projects accurately. Steve understands that apart from generating reports, the *Project Tracking System (PTS)* should also provide users with options that Stella is asking for. So, he plans to create a report (which includes the details of existing employees) with a form (to create new employee entries). Also, as there are many new projects coming in, Steve creates a form, which Stella can use to manage new projects.

Let's see how Steve meets these requirements.



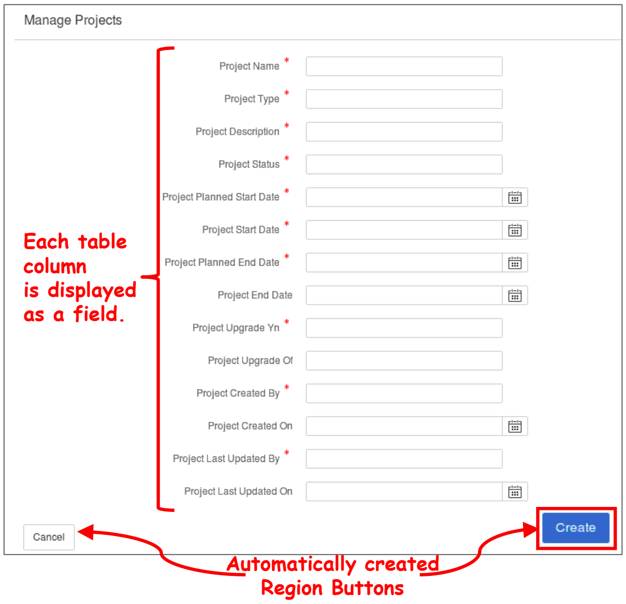
Lesson Agenda

- Forms: Introduction
  - What Are Forms?
  - Types of Forms
- Creating Forms
- Modifying Forms

What are forms?

Forms: Introduction

- What are forms in Oracle Application Express?
  - Forms are application components that are used to manipulate database tables and objects.
- How are forms created in Oracle Application Express?
  - Manually
  - Declaratively by using wizards
- Where are forms created?
  - On a new page in the application
  - On an existing page in the application



Manage Projects

Project Name *
Project Type *
Project Description *
Project Status *
Project Planned Start Date *
Project Start Date *
Project Planned End Date *
Project End Date
Project Upgrade Yn *
Project Upgrade Of
Project Created By *
Project Created On
Project Last Updated By *
Project Last Updated On

Each table column is displayed as a field.

Automatically created Region Buttons

The *PTS* application has a user interface using which users can enter project details, such as the type of project, project status, and so on. Oracle Application Express supports forms to enable this type of user interface. Let's explore and see what forms are and what they will enable you to do.

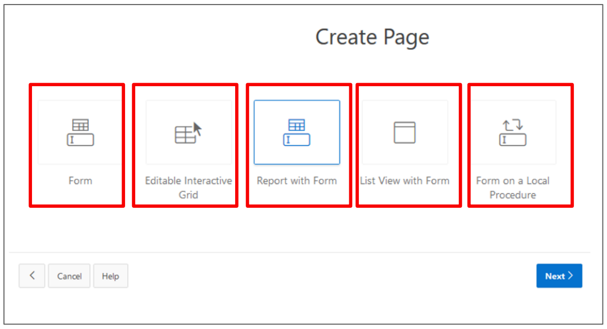
Forms are application components that take input from a user and submit it to a server. A form usually consists of one or more page items (drop-down list, text box, check box, option buttons, and so on), which enable users to enter information, and a button or link with a submit action.

In Oracle Application Express, you use forms when you must gather input from a user before performing a task on a database table. For example, you can create a form to insert data into a database table.

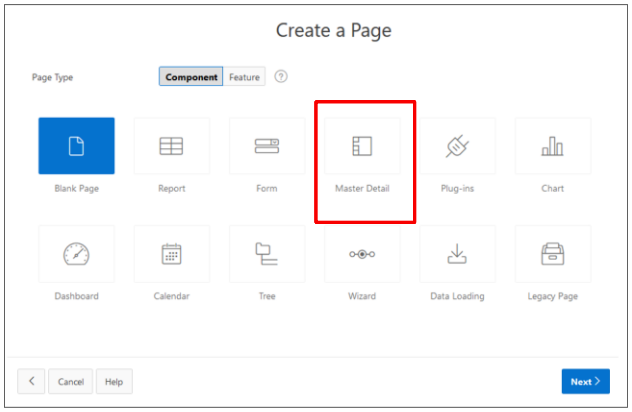
In Oracle Application Express, you can create forms easily by using wizards. The wizard automatically includes the necessary buttons and processes that are required to create, update, and delete rows from the table. In this slide, you can see a sample form that you will create later in this lesson.

Types of Forms

Create Form Wizard



Create Page Wizard



Oracle Application Express provides wizards to create multiple types of forms. See Oracle Application Express Documentation on *Forms* (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/developing-forms.html#GUID-FDB1159B-21B7-47BE-8640-2C92CCE2389E).

In this lesson, you learn how to create forms by using the following wizards:

- Form wizard: You use this form to update a single row on the specified data source, such as a table, SQL Query, or Web Source Module.
- Report with Form wizard: You use this form to create a report and form on a table you select. You select the report type (Interactive Report, Interactive Grid, or Classic Report). Each row in the report provides a link to the form to enable users to update each record.

Note: This wizard does not support tables having more than 127 columns. Selecting more than 127 columns generates an error.

- List View with Form wizard: You use this form mainly for mobile applications. It creates a form and list view that enables users to update a single entry in a database table. You choose the table on which to build the form and select the column to be used for displaying text in List View.
- Form on a Local Procedure wizard: You use this wizard to create forms based on stored procedure arguments. Use this approach when you have implemented logic or Data Manipulation Language (DML) in a stored procedure or package.
- Master Detail wizard: You use this form to display a master row and multiple detail rows. When you create a master detail form using the Create Page or Create Application Wizards, you choose the tables on which to build the master form and the detail form. Master detail form options include: *Stacked*, *Side by Side*, and *Drill Down*.

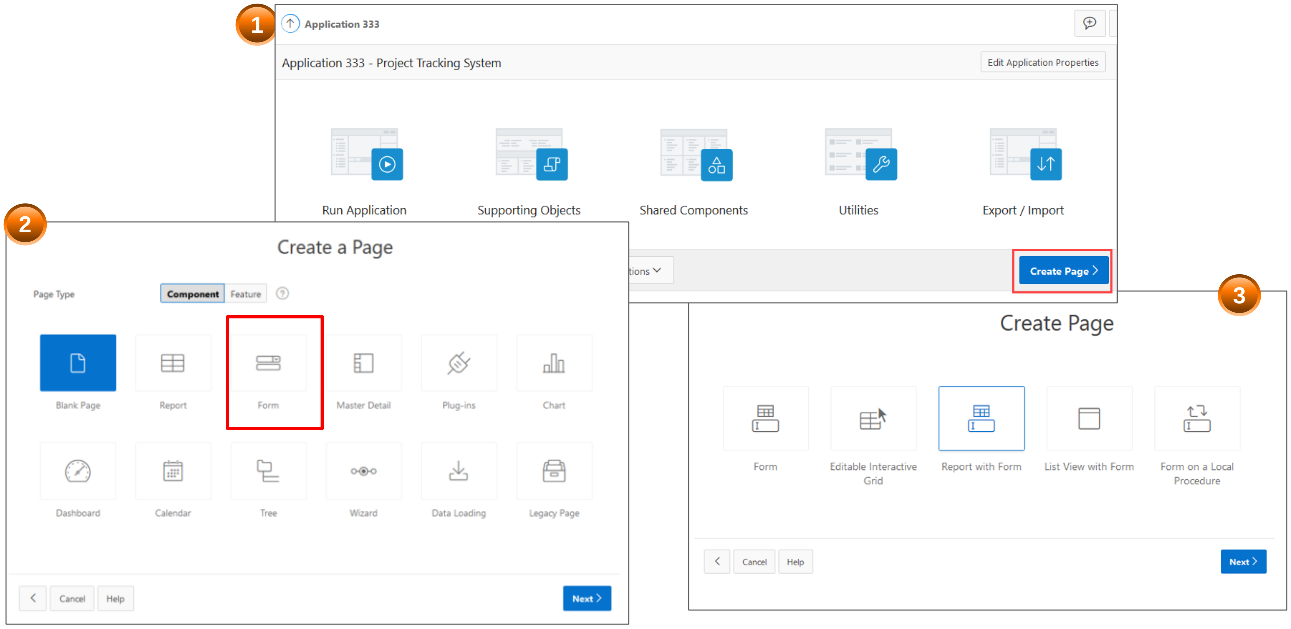
You will learn how to access them in the next few slides.

Lesson Agenda

- Introducing Forms
- Creating Forms
  - Accessing the Create Form Wizard
  - ROWID Versus Primary Key
  - Creating a Form
  - Adding a Form as a Region
  - Creating Report with Form
  - Creating a Form with a List View
  - Creating a Stacked Master Detail Form
- Modifying Forms
- Creating Forms in a Mobile Application

ORACLE

Accessing the Create Form Wizards



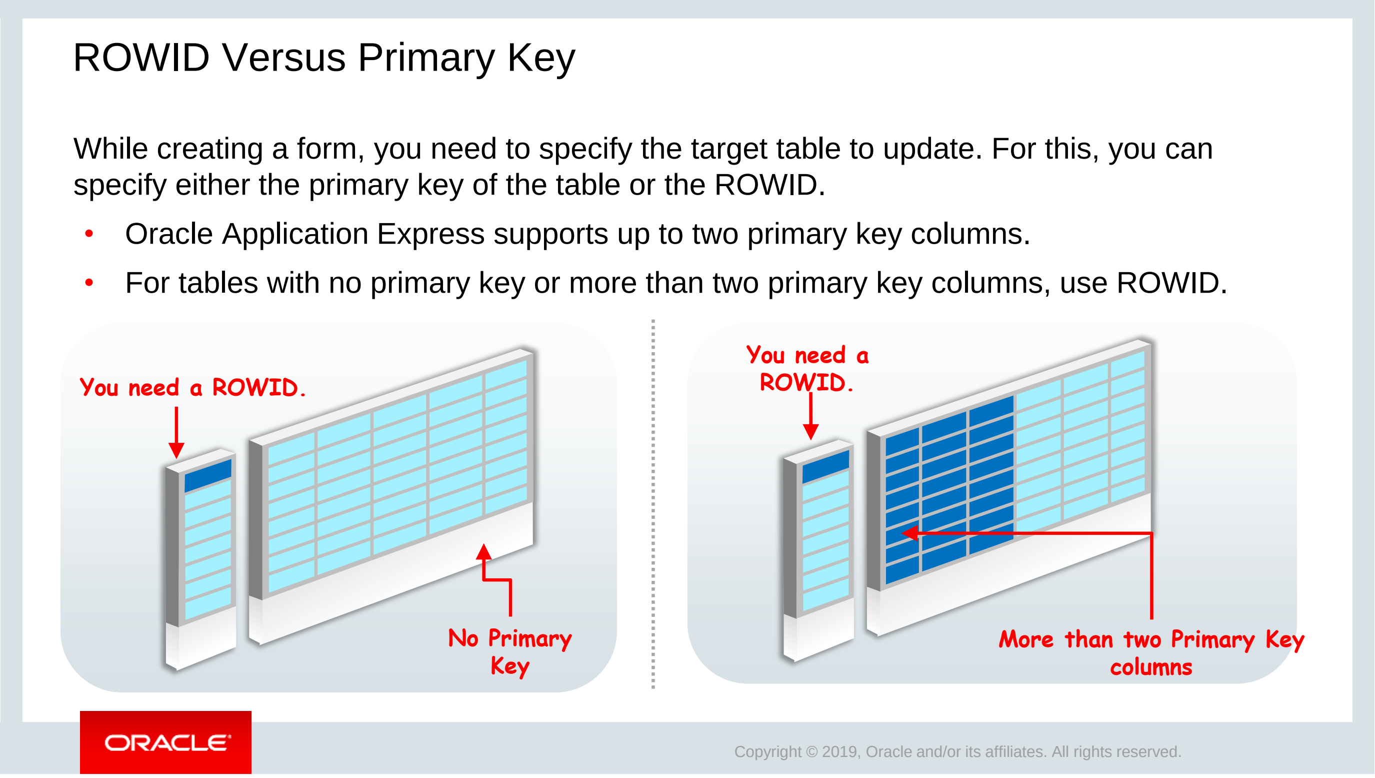
ORACLE

To access the Create Form Wizards while creating a new page (for example, *Form*) in your application, perform the following steps:

1. Click **Create Page** on the home page of the application where you want to create the form (screenshot 1).
2. The Create Page Wizard opens. Select **Form** from the available options and click **Next** (screenshot 2).
3. The form wizards are displayed. You can select a wizard based on the type of form that you want to create, such as Form, Editable Interactive Grid, Report with Form, List View with Form, and Form on a Local Procedure (screenshot 3).

You can also access these wizards while creating a region on a page. For more information about regions, refer to the lesson titled “Working with Pages and Regions.”

Note: To create a *master detail* form, you have to use the *Create Page Wizard*. You will learn about this later in this lesson.



ROWID Versus Primary Key

While creating a form, you need to specify the target table to update. For this, you can specify either the primary key of the table or the ROWID.

- Oracle Application Express supports up to two primary key columns.
- For tables with no primary key or more than two primary key columns, use ROWID.

You need a ROWID.

No Primary Key

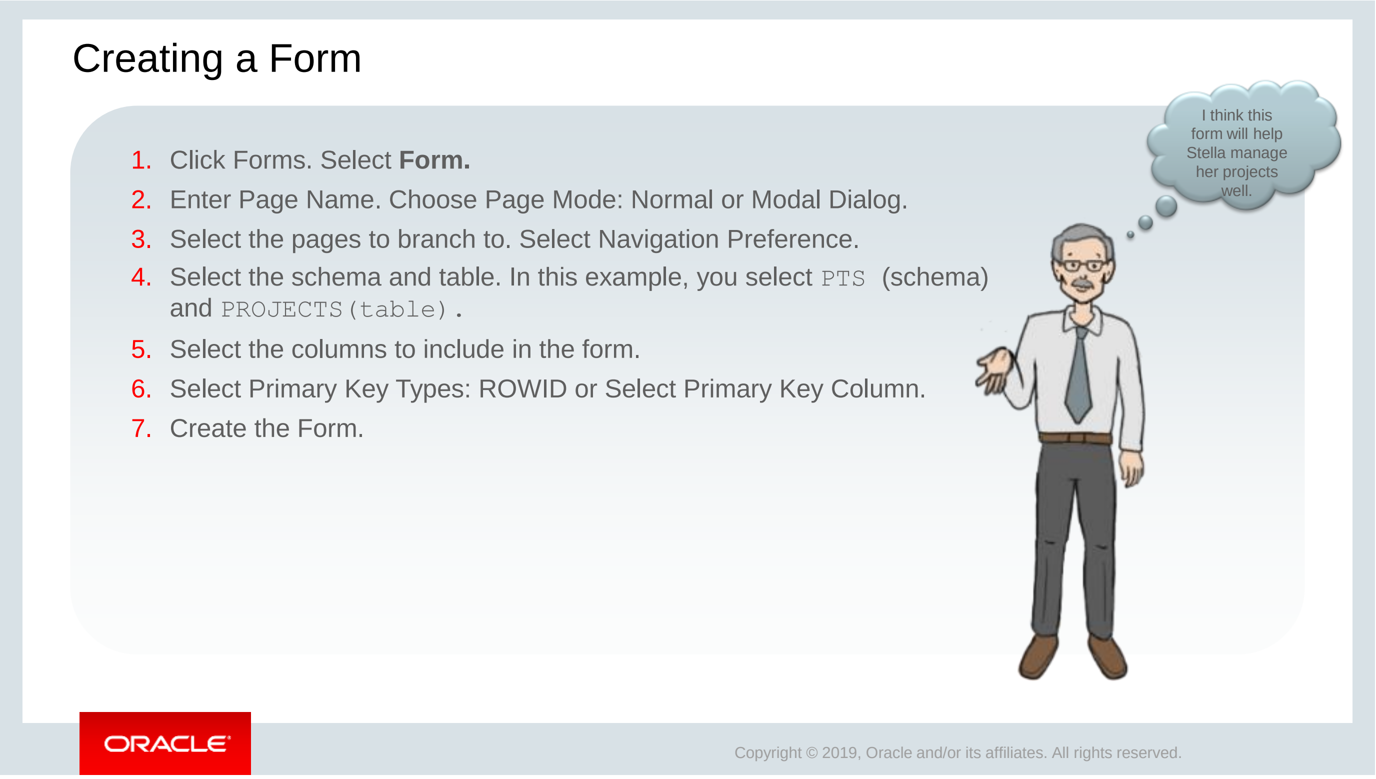
You need a ROWID.

More than two Primary Key columns

Before you start creating a form, it's important to note the difference between a **ROWID** and a **Primary Key**.

The most common practice is to specify a primary key for the table. A primary key can be a single column or a combination of two or more columns in a table. In Oracle Application Express, the Create Form wizards allow you to specify up to a maximum of two columns for a primary key.

If your table does not have a primary key or if it has three or more primary key columns, Oracle Application Express recommends that you use the ROWID feature. ROWID is a pseudocolumn that uniquely identifies a row in a table.



Creating a Form

1. Click Forms. Select **Form.**
2. Enter Page Name. Choose Page Mode: Normal or Modal Dialog.
3. Select the pages to branch to. Select Navigation Preference.
4. Select the schema and table. In this example, you select `PTS` (schema) and `PROJECTS(table)`.
5. Select the columns to include in the form.
6. Select Primary Key Types: ROWID or Select Primary Key Column.
7. Create the Form.

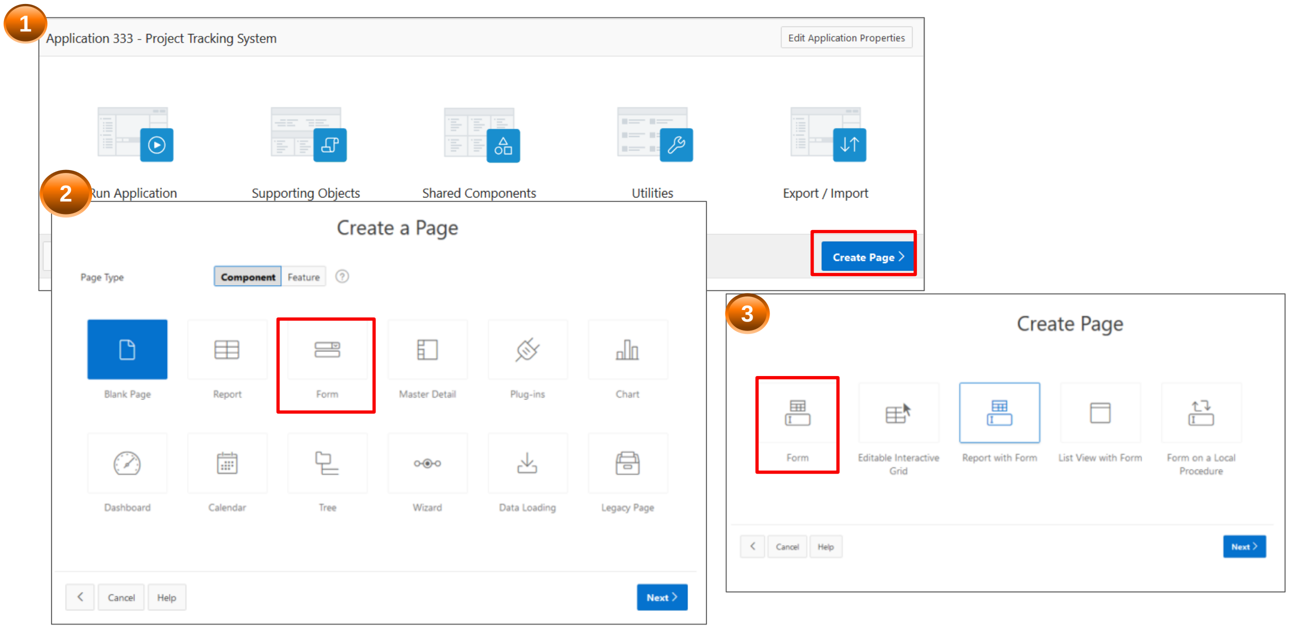
I think this form will help Stella manage her projects well.

Remember that Stella wanted help to update her projects and create new projects. Steve, therefore, uses this wizard to create a Projects Form for Stella to manage her projects. Using this form, she can create a new project or update an existing project. This form will provide all details related to the type of project, project start date, project end date, and so on.

In step 3, Steve sets the Branch on Submit and Cancel to Page 3, which is the *Projects Master Report*. Therefore, every time Stella creates a new form or cancels a form entry, she is redirected to that page. You will learn more about Branching in the lesson titled "Adding Page Processing."

In step 4, because Steve is creating a *Manage Projects* Form for Stella, he selects `PROJECTS (table)`.

Example: Creating a Form

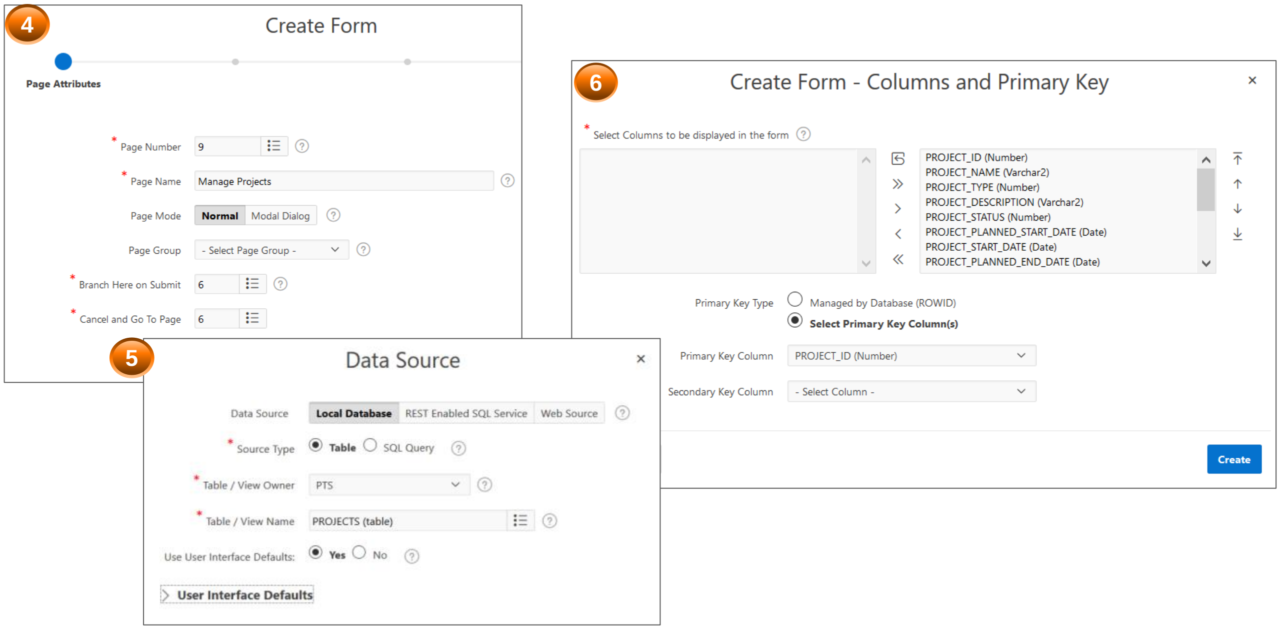


The *PTS* application requires a user interface for managers to enter project information. In an earlier lesson, Steve had created *Projects Master Report*, which presents complete details of all the projects managed using the *PTS* application. This report allows project managers to manipulate the report and see only those details that they are interested in and for only those projects which they manage.

But Steve has not yet created any component that allows users to enter new Projects into PTS. So, he now creates *Form on a Table* using Oracle Application Express for the `PROJECTS` table. Let’s see how he does it:

1. Click Create Page from your application home page.
2. Select Form (screenshot 2). In the Create Form Wizard, again select Form and click Next (screenshot 3).

Example: Creating a Form



Create Form

Page Attributes

* Page Number: 9

* Page Name: Manage Projects

Page Mode: **Normal** | Modal Dialog

Page Group: - Select Page Group -

* Branch Here on Submit: 6

* Cancel and Go To Page: 6

Data Source

Data Source: **Local Database** | REST Enabled SQL Service | Web Source

* Source Type: ● Table ○ SQL Query

* Table / View Owner: PTS

* Table / View Name: PROJECTS (table)

Use User Interface Defaults: ● Yes ○ No

User Interface Defaults

Create Form - Columns and Primary Key

* Select Columns to be displayed in the form

PROJECT_ID (Number)
PROJECT_NAME (Varchar2)
PROJECT_TYPE (Number)
PROJECT_DESCRIPTION (Varchar2)
PROJECT_STATUS (Number)
PROJECT_PLANNED_START_DATE (Date)
PROJECT_START_DATE (Date)
PROJECT_PLANNED_END_DATE (Date)

Primary Key Type: ○ Managed by Database (ROWID) ● Select Primary Key Column(s)

Primary Key Column: PROJECT_ID (Number)

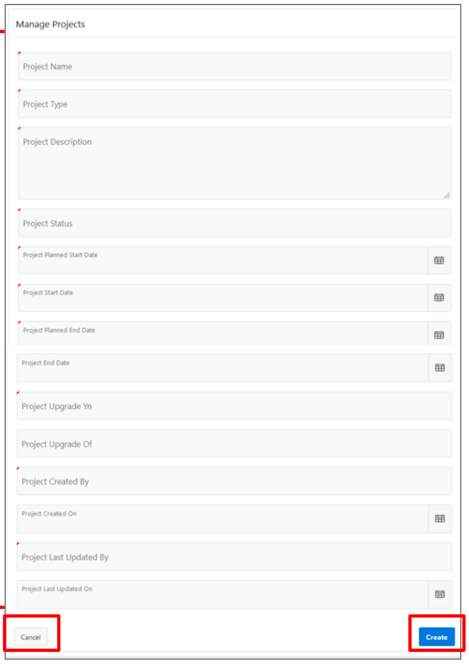
Secondary Key Column: - Select Column -

Create

3. Enter the following values. Leave the remaining as default and click Next (screenshot 4).
   - **Page Number**: Enter `9`.
   - **Page Name**: Enter `Manage Projects`.
   - **Branch here on Submit**: Enter `6` (*Projects Master Report*).
   - **Cancel and Go To Page**: Enter `6` (*Projects Master Report*).
4. Accept the default for Navigation Preference and click Next.
5. Enter the following values and click Next (screenshot 5).
   - **Table/View Owner**: Select `PTS`
   - **Table/View Name**: Select `PROJECTS (table).` Notice that all the columns are automatically selected for display.
   - **Use User Interface Defaults:** `Yes` (default)
6. Enter the following values in the *Create Form on Table* page and click Create (screenshot 6).
   - **Primary Key Type**: Select Select `Primary Key Columns(s).`
   - **Primary Key Column**: Select `PROJECT_ ID(Number)`

The *Manage Projects* form will open in Page Designer view. Click **Save and Run** to view the Form.

Example: Creating a Form



Manage Projects

Project Name

Project Type

Project Description

Project Status

Project Planned Start Date

Project Start Date

Project Planned End Date

Project End Date

Project Upgrade Yn

Project Upgrade Of

Project Created By

Project Created On

Project Last Updated By

Project Last Updated On

Each table column is displayed as a field.

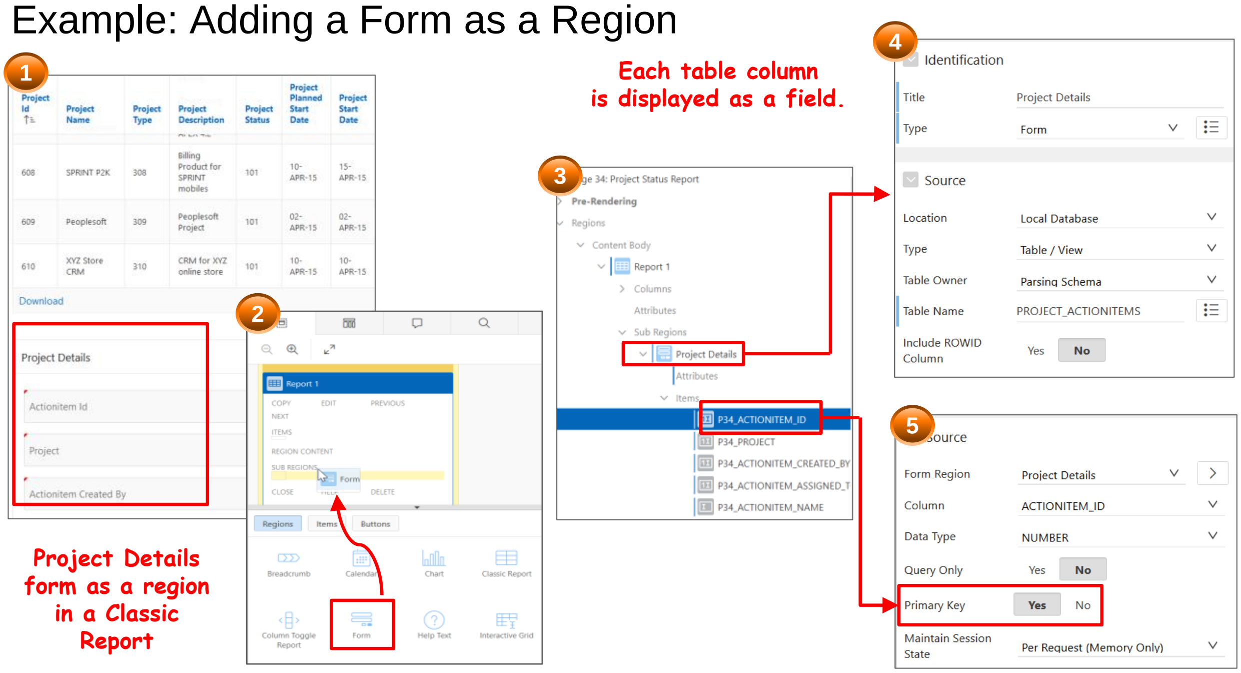
Automatically created Region Buttons

Steve now has a report and a form to manage projects separately as *Projects Master Report* and *Manage Projects* form. Later in this lesson, he would establish a link between this report and form.

The wizard creates a form where users can enter values for the selected columns of the *PROJECTS* table. The wizard automatically creates two buttons on the Form: **Cancel** and **Create**.

To create a new project in the table, Stella can enter the details and click the **Create** button. The data is inserted in the table, and Stella is redirected to the page that Steve specified while creating the form (that is Page 6, *Projects Master Report*). **Cancel** returns her to Page 6 too because Steve had specified that while creating the form.

Example: Adding a Form as a Region



1
2
3
4
5
Each table column is displayed as a field.
Project Details form as a region in a Classic Report

1

Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date
608	SPRINT P2K	308	Billing Product for SPRINT mobiles	101	10-APR-15	15-APR-15
609	Peoplesoft	309	Peoplesoft Project	101	02-APR-15	02-APR-15
610	XYZ Store CRM	310	CRM for XYZ online store	101	10-APR-15	10-APR-15

Download

Project Details

Actionitem Id

Project

Actionitem Created By

4

Identification

Title	Project Details
Type	Form

Source

Location	Local Database
Type	Table / View
Table Owner	Parsing Schema
Table Name	PROJECT_ACTIONITEMS
Include ROWID Column	Yes [✓ No]

5

Source

Form Region	Project Details
Column	ACTIONITEM_ID
Data Type	NUMBER
Query Only	Yes [✓ No]
Primary Key	[✓ Yes] No
Maintain Session State	Per Request (Memory Only)

ORACLE

In the previous example, you can see that the form is created as a separate page in the application. You also have the option to add a form as a region in a report.

Screenshot 1 in the slide shows the *Project Details* form as a region of a report. To add a form as a region, you must drag the form from the Region tab to the report subregion (screenshot 2) and then define the form properties (screenshot 4 and 5). Most important, you must select and define the primary key in the form. In this example, `ACTIONITEM_ID` is defined as the primary key, as shown in screenshot 5.

Note that in the forms that are created using a wizard, the Cancel and Create buttons are created automatically. But if you add a form as a region, you must add the Create and Cancel buttons manually.

Creating a Report with Form

1. Click Forms. Select “Report with Form.”
2. Select the Report Type (Interactive Report or Interactive Grid or Classic Report).
3. Enter Report and Form Page Name and Numbers.
4. Choose Page Mode: Normal or Modal Dialog.
5. Select Navigation Preference.
6. Select the schema and table. In this example, you select PTS (schema) and EMPLOYEES(table).
7. Select the columns to include in the form.
8. Select Primary Key Types: ROWID or Select Primary Key Column.
9. Create the Form.

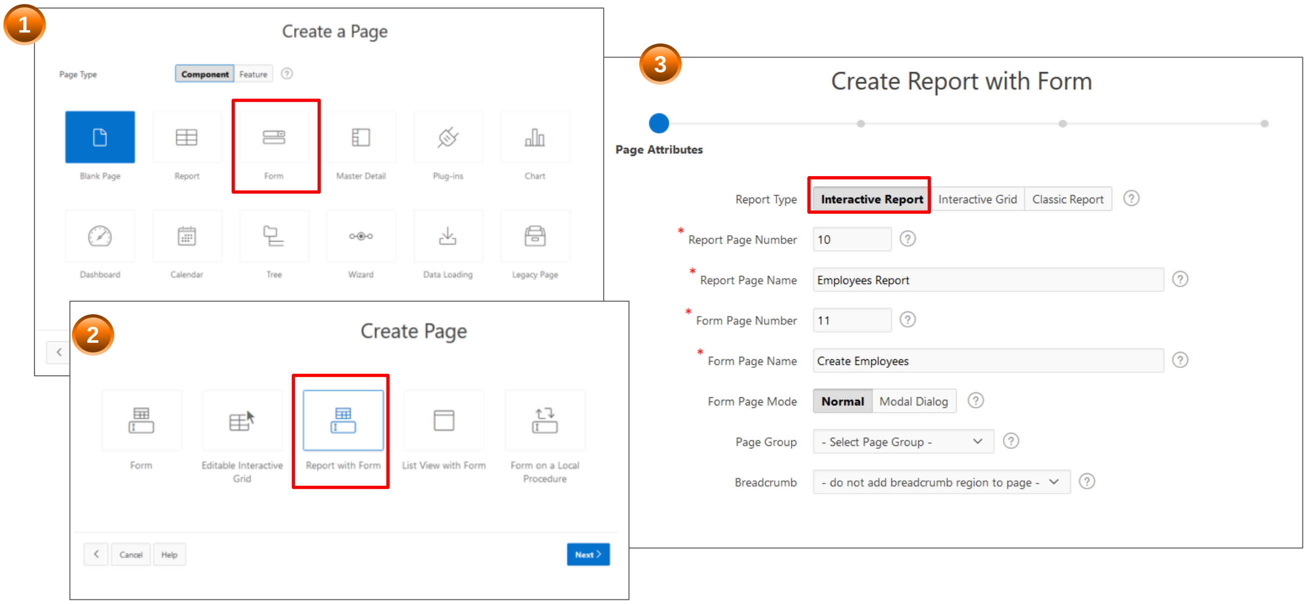
This looks great! I can now create new entries and update my Employees table with ease.

You created a *Report with Form on Table* while creating an Interactive Grid. In this topic, you will revisit this by using the Forms page.

Steve now tries to find a solution for Stella's next requirement of having an updated employees table, so that she can assign projects to them accurately. For that, he wants to create an *Employees Report* along with a *Create Employees* form for Stella. He uses the Report with Form on Table to help Stella update her *Employees Report* (for example, phone number or address of existing employees). Also, each time a new employee joins, Stella can click the *Create Employees* form to create a new employee record with the details of the new employee.

Because Steve wants to help Stella update her *Employees Report* and also create a new entry for each new employee, he selects `EMPLOYEES(table)` in step 6.

Example: Creating a Report with Form



Report Type: Interactive Report | Interactive Grid | Classic Report

* Report Page Number: 10

* Report Page Name: Employees Report

* Form Page Number: 11

* Form Page Name: Create Employees

Form Page Mode: Normal | Modal Dialog

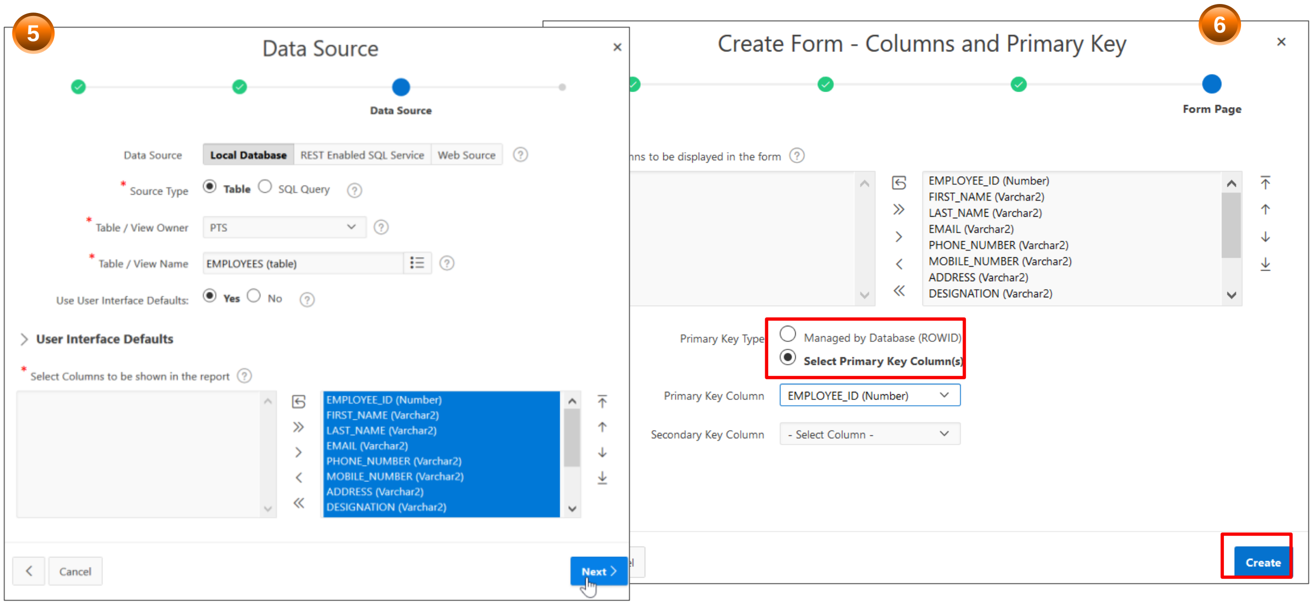
Page Group: - Select Page Group -

Breadcrumb: - do not add breadcrumb region to page -

Project Managers who are using *PTS* for managing their projects will also need an interface to add or modify the employees who work in that organization. Only when the employees database is up-to-date, they can be allocated to the various projects accurately. So, Steve starts creating a form on the `EMPLOYEES` table with Report, which will give them a preview of the existing employee details and also the option to create a new employee entry. Let's see how.

1. Click Create Page from your application home page.
2. Select Form (screenshot 1).
3. Select Report with Form (screenshot 2).
4. Select the report type: Interactive Report, Interactive Grid, or Classic Report. In this example, Steve selects Interactive Report. Enter the following values. Leave the remaining as default and click Next (screenshot 3).
   - **Report Page Number**: Enter `10`.
   - **Report Page Name**: Enter `Employees Report`.
   - **Form Page Number**: Enter `11`.
   - **Form Page Name**: Enter `Create Employees`.
5. Accept the Navigation Preference - Here you choose Create a new navigation menu entry) and click Next.

Example: Creating a Report with Form



5

Data Source

Data Source

Data Source: **Local Database** | REST Enabled SQL Service | Web Source

* Source Type ● **Table** ○ SQL Query

* Table / View Owner: PTS

* Table / View Name: EMPLOYEES (table)

Use User Interface Defaults: ● **Yes** ○ No

> **User Interface Defaults**

* Select Columns to be shown in the report

EMPLOYEE_ID (Number)
FIRST_NAME (Varchar2)
LAST_NAME (Varchar2)
EMAIL (Varchar2)
PHONE_NUMBER (Varchar2)
MOBILE_NUMBER (Varchar2)
ADDRESS (Varchar2)
DESIGNATION (Varchar2)

6

Create Form - Columns and Primary Key

Form Page

...ns to be displayed in the form

EMPLOYEE_ID (Number)
FIRST_NAME (Varchar2)
LAST_NAME (Varchar2)
EMAIL (Varchar2)
PHONE_NUMBER (Varchar2)
MOBILE_NUMBER (Varchar2)
ADDRESS (Varchar2)
DESIGNATION (Varchar2)

Primary Key Type: ○ Managed by Database (ROWID) ● **Select Primary Key Column(s)**

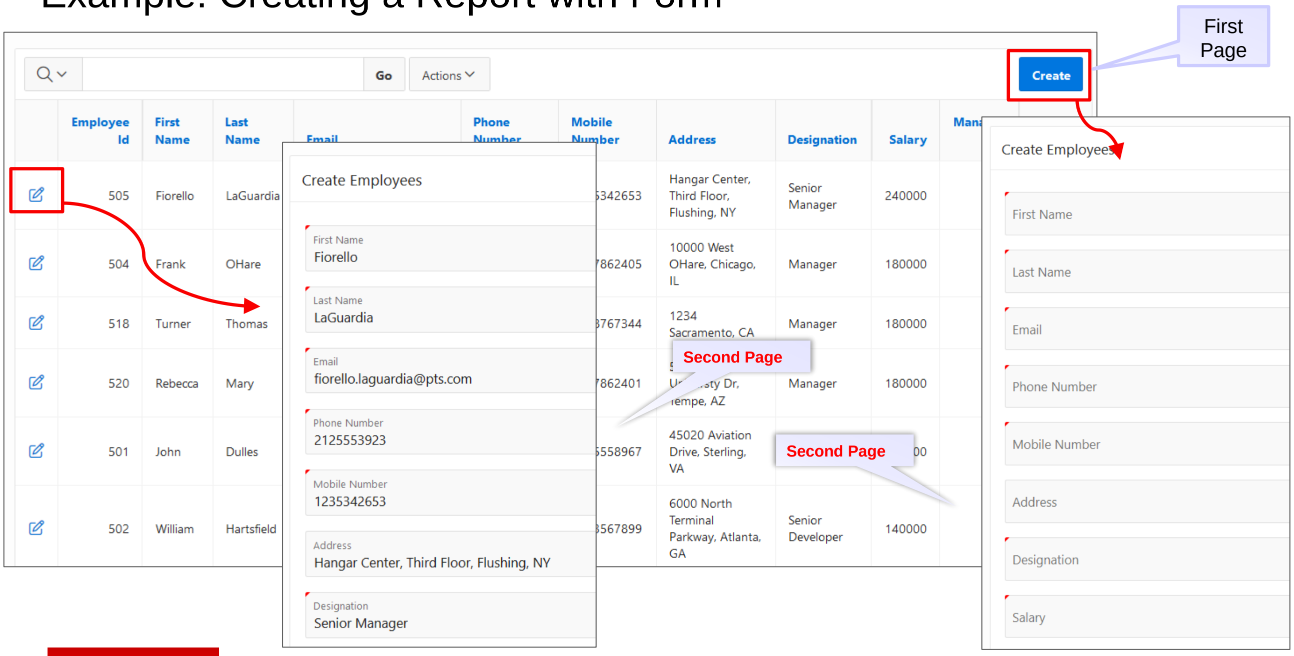
Primary Key Column: EMPLOYEE_ID (Number)

Secondary Key Column: - Select Column -

5. Enter the following values and click Next (screenshot 5).
   - **Table/View Owner**: Select `PTS`
   - **Table/View Name**: Select `EMPLOYEES (table).` Notice that all the columns are automatically selected for display.
   - **Use User Interface Defaults:** Select **Yes** (default)
6. Select all columns and move them to the box on the right to include all of them in the form or simply click >>. Enter the following values and click Create (screenshot 6).
   - **Primary Key Type**: Select **Primary Key Columns(s).**
   - **Primary Key Column 1**: Select `EMPLOYEE_ID.` Leave the **Secondary Key Column** blank.

The *Employees Report* will open in Page Designer view. Click the Run icon to view the report.

Example: Creating a Report with Form



	Employee Id	First Name	Last Name	Email	Phone Number	Mobile Number	Address	Designation	Salary	Mana
✎	505	Fiorello	LaGuardia			342653	Hangar Center, Third Floor, Flushing, NY	Senior Manager	240000	
✎	504	Frank	OHare			7862405	10000 West OHare, Chicago, IL	Manager	180000	
✎	518	Turner	Thomas			3767344	1234 Sacramento, CA	Manager	180000	
✎	520	Rebecca	Mary			7862401	sty Dr, empe, AZ	Manager	180000	
✎	501	John	Dulles			5558967	45020 Aviation Drive, Sterling, VA		00	
✎	502	William	Hartsfield			3567899	6000 North Terminal Parkway, Atlanta, GA	Senior Developer	140000	

Create Employees

First Name	Fiorello
Last Name	LaGuardia
Email	fiorello.laguardia@pts.com
Phone Number	2125553923
Mobile Number	1235342653
Address	Hangar Center, Third Floor, Flushing, NY
Designation	Senior Manager

Create Employees

First Name

Last Name

Email

Phone Number

Mobile Number

Address

Designation

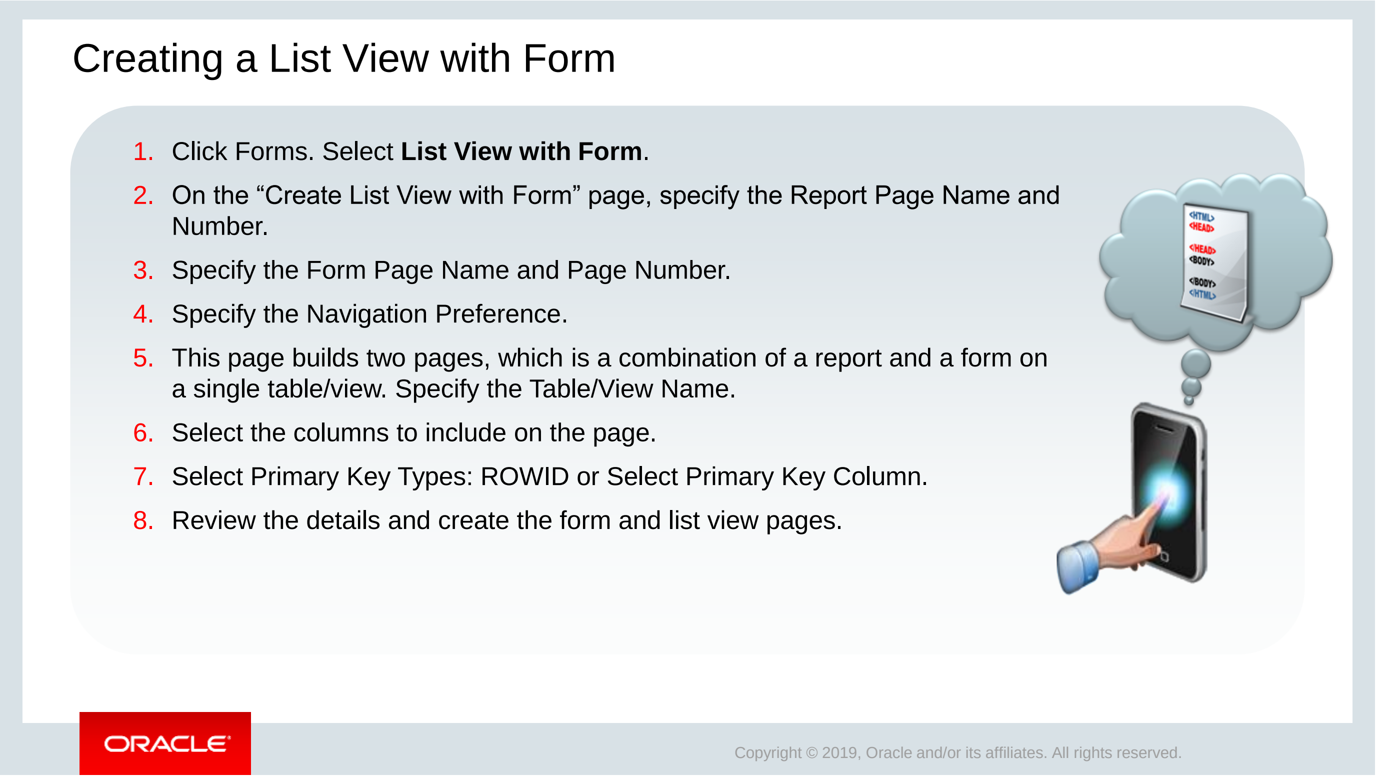
Salary

ORACLE

Report with Form combines the steps to create a report and the steps to create a form—and creates two pages. The first page is a report with an edit link (link column) for each row. The report page also includes a **Create** button to enable users to insert new entries (for example, new employee details) into the table. The second page is a form to create, edit, or delete the row selected from the first page (reports page).

As you see in this slide, two pages are created by using the *Report with Form*. The first page is an interactive report that lists the details from an `EMPLOYEES` table. When you click the **Create** button, a form appears where you can insert new employee details in the `EMPLOYEES` table.

When you click the Edit icon in the report, the form is populated with the row details. You can edit the details and save your changes.



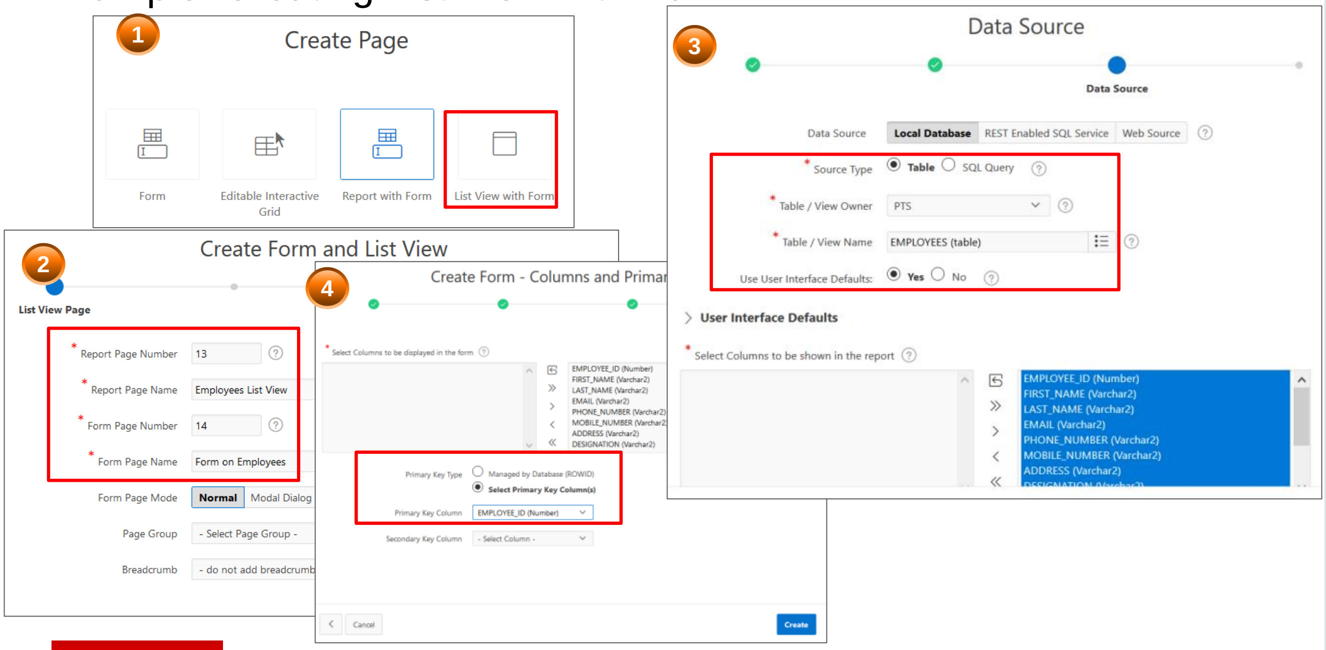
Creating a List View with Form

1. Click Forms. Select **List View with Form**.
2. On the “Create List View with Form” page, specify the Report Page Name and Number.
3. Specify the Form Page Name and Page Number.
4. Specify the Navigation Preference.
5. This page builds two pages, which is a combination of a report and a form on a single table/view. Specify the Table/View Name.
6. Select the columns to include on the page.
7. Select Primary Key Types: ROWID or Select Primary Key Column.
8. Review the details and create the form and list view pages.

Now, Steve is wondering what if Stella wants to access her forms via her mobile device. Can she do it? And that is when he thought about the *Report with a List View on Table*. Mostly a report will show a grid of rows and columns with information. But on a smartphone screen, room to show information is limited. For this purpose, a list view is ideal.

The slide above provides an overview of the steps to create a form by using the *Report with a List View on Table*. You must access the *Report with a List View on Table* and follow the instructions.

Example: Creating List View with Form



1 Create Page

Form | Editable Interactive Grid | Report with Form | List View with Form

2 Create Form and List View

List View Page

* Report Page Number: 13

* Report Page Name: Employees List View

* Form Page Number: 14

* Form Page Name: Form on Employees

Form Page Mode: **Normal** | Modal Dialog

Page Group: - Select Page Group -

Breadcrumb: - do not add breadcrumb

3 Data Source

Data Source

Data Source: **Local Database** | REST Enabled SQL Service | Web Source

* Source Type: ● **Table** ○ SQL Query

* Table / View Owner: PTS

* Table / View Name: EMPLOYEES (table)

Use User Interface Defaults: ● **Yes** ○ No

User Interface Defaults

* Select Columns to be shown in the report

EMPLOYEE_ID (Number)
FIRST_NAME (Varchar2)
LAST_NAME (Varchar2)
EMAIL (Varchar2)
PHONE_NUMBER (Varchar2)
MOBILE_NUMBER (Varchar2)
ADDRESS (Varchar2)
DESIGNATION (Varchar2)

4 Create Form - Columns and Primar

* Select Columns to be displayed in the form

EMPLOYEE_ID (Number)
FIRST_NAME (Varchar2)
LAST_NAME (Varchar2)
EMAIL (Varchar2)
PHONE_NUMBER (Varchar2)
MOBILE_NUMBER (Varchar2)
ADDRESS (Varchar2)
DESIGNATION (Varchar2)

Primary Key Type: ○ Managed by Database (ROWID) ● **Select Primary Key Column(s)**

Primary Key Column: EMPLOYEE_ID (Number)

Secondary Key Column: - Select Column -

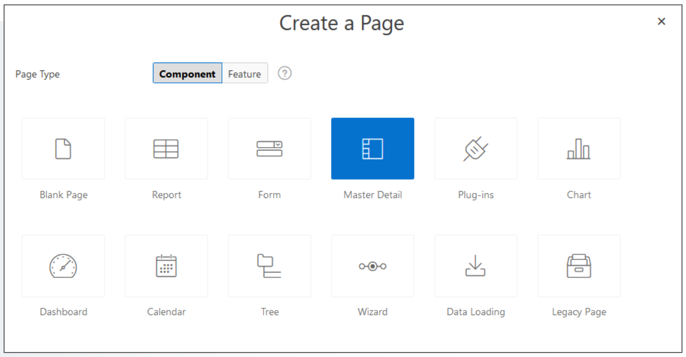
In this slide, Steve creates a form on EMPLOYEES table with a List View. EMPLOYEE_ID is displayed in the list view, and Stella can easily create or modify an Employee record by using this Form via her mobile devices. To create a list view form:

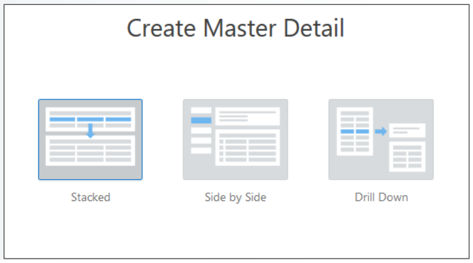
Select **Form** and **List View with Form** (screenshot 1).

1. Enter the following values and click Next (screenshot 2).
   - **Report Page Number**: Enter `13`.
   - **Report Page Name**: Enter `Employees List View`.
   - **Form Page Number**: Enter `14`.
   - **Form Page Name**: Enter `Form on Employees`.
2. Select Create a new navigation menu entry and click Next after entering the following values:
   - **New Navigation Menu Entry**: Select *Employees List View.*
   - **Parent Navigation Menu Entry**: Select *Home.*
3. Select `EMPLOYEES` from the Table/View Name drop-down list. Click Next. Select all columns and move them to the box on the right to include all of them in the form (screenshot 3).
4. For Primary Key Type: Select Select Primary Key Columns(s). Select `EMPLOYEE_ID` for Primary Key Column and click Create (screenshot 4). Leave the Secondary Key Column blank.
5. The *Employees List View* report will open in Page Designer view. Click the Run icon to view the list view report.
6. Click any row from the list view report, and the form opens for updating the record. Click Cancel to go back to the list view. Click the Create button provided on the report to add a new Employee.

Creating a Stacked Master Detail Form

1. Select Master Detail from the **Create a Page** Wizard.
2. Choose the type of master detail form you want to create. In this example, you select Stacked.
3. Enter Master Region Title, Table/View Owner, Table/View Name, Primary Key Column. In this example, you select `PTS` (schema) and `PROJECT_TYPES(table)`.
4. Enter Detail Region Title, Table/View Owner, Table/View Name, Primary Key Column, and Master Detail Foreign Key. In this example, you select `PTS` (schema) and `PROJECTS(table)`.
5. Create the form.





Now, because there are many new and existing projects, Steve wants to help his managers manage the different project types. For that, he plans to create a master detail form on PROJECT_TYPES and PROJECTS tables.

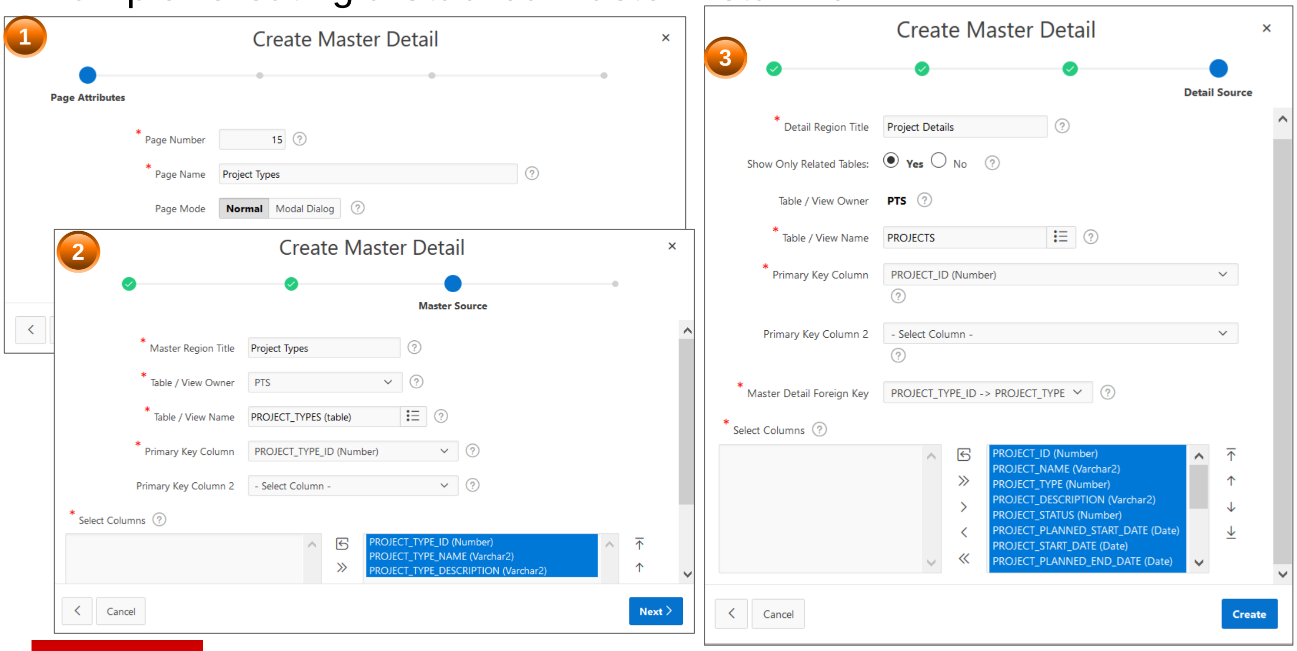
A master detail form reflects a one-to-many relationship between two tables in a database. These forms enable users to insert, update, and delete values from two tables or views. You choose the tables on which to build the master and detail regions. There are three types of master detail forms: **Stacked**, **Drill Down**, and **Side by Side**.

To create a **master detail** form:

1. Go to your application and click **Add Page**.
2. Select **Master Detail**.
3. Choose the type of master detail form that you would like to create.

The slide provides an overview of the steps to create a *Stacked* master detail form. You must access the wizard and follow the wizard instructions.

Example: Creating a Stacked Master Detail Form



1 Create Master Detail

Page Attributes

* Page Number: 15

* Page Name: Project Types

Page Mode: Normal | Modal Dialog

2 Create Master Detail

Master Source

* Master Region Title: Project Types

* Table / View Owner: PTS

* Table / View Name: PROJECT_TYPES (table)

* Primary Key Column: PROJECT_TYPE_ID (Number)

Primary Key Column 2: - Select Column -

* Select Columns:

PROJECT_TYPE_ID (Number)
PROJECT_TYPE_NAME (Varchar2)
PROJECT_TYPE_DESCRIPTION (Varchar2)

3 Create Master Detail

Detail Source

* Detail Region Title: Project Details

Show Only Related Tables: ● Yes ○ No

Table / View Owner: PTS

* Table / View Name: PROJECTS

* Primary Key Column: PROJECT_ID (Number)

Primary Key Column 2: - Select Column -

* Master Detail Foreign Key: PROJECT_TYPE_ID -> PROJECT_TYPE

* Select Columns:

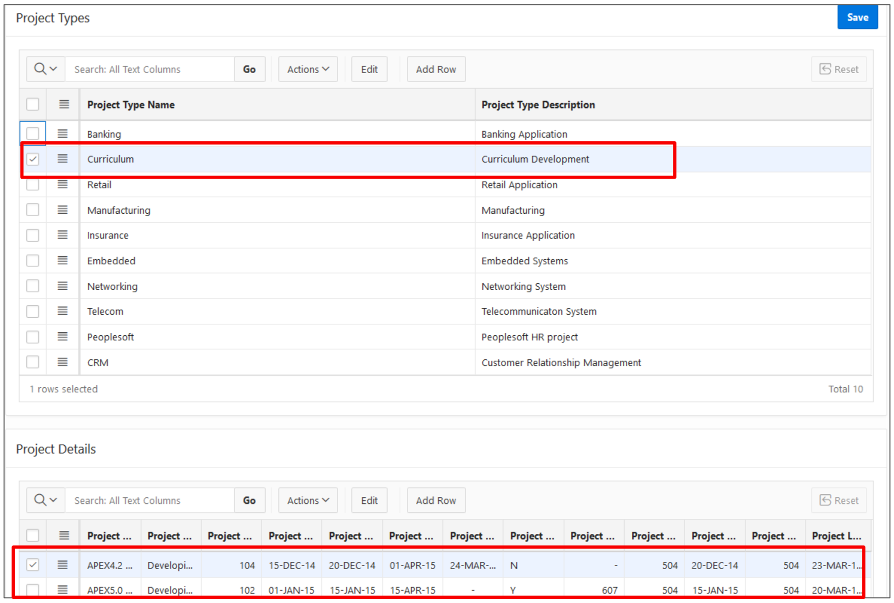
PROJECT_ID (Number)
PROJECT_NAME (Varchar2)
PROJECT_TYPE (Number)
PROJECT_DESCRIPTION (Varchar2)
PROJECT_STATUS (Number)
PROJECT_PLANNED_START_DATE (Date)
PROJECT_START_DATE (Date)
PROJECT_PLANNED_END_DATE (Date)

Let's see how Steve creates the *Stacked master detail* form:

1. Select Master Detail Form and then select Stacked (see slide 22).
2. Enter the following values. Leave the remaining as default and click Next (screenshot 1):
   - **Page Number:** Enter `15`.
   - **Page Name:** Enter `Project Types`.
3. Accept the default navigation menu.
4. Enter the following values (screenshot 2):
   - **Master Region Title**: Enter `Project Types`
   - **Table/View Owner**: Select `PTS`
   - **Table/View Name**: Select `PROJECT_TYPES(table)`
   - **Primary Key Column**: Select `PROJECT_TYPE_ID(Number)`
5. Enter the following values (screenshot 2):
   - **Detail Region Title:** Enter `Project Details`.
   - **Table/View Owner:** Select `PTS`
   - **Table/View Name:** Select `PROJECTS`
   - **Primary Key Column:** Select `PROJECT_ID`
   - **Master Detail Foreign Key**: Select `PROJECT_TYPE_ID->PROJECT_TYPE`
6. Click Create.
7. The *Project Types* page will open in Page Designer view. Click the Run icon.

In the following slide, you can see both *Project Types* and *Project Details* reports displayed on the same page.

Example: Creating a Stacked Master Detail



Project Types

□	≡	Project Type Name	Project Type Description
□	≡	Banking	Banking Application
☑	≡	Curriculum	Curriculum Development
□	≡	Retail	Retail Application
□	≡	Manufacturing	Manufacturing
□	≡	Insurance	Insurance Application
□	≡	Embedded	Embedded Systems
□	≡	Networking	Networking System
□	≡	Telecom	Telecommunicaton System
□	≡	Peoplesoft	Peoplesoft HR project
□	≡	CRM	Customer Relationship Management

1 rows selected Total 10

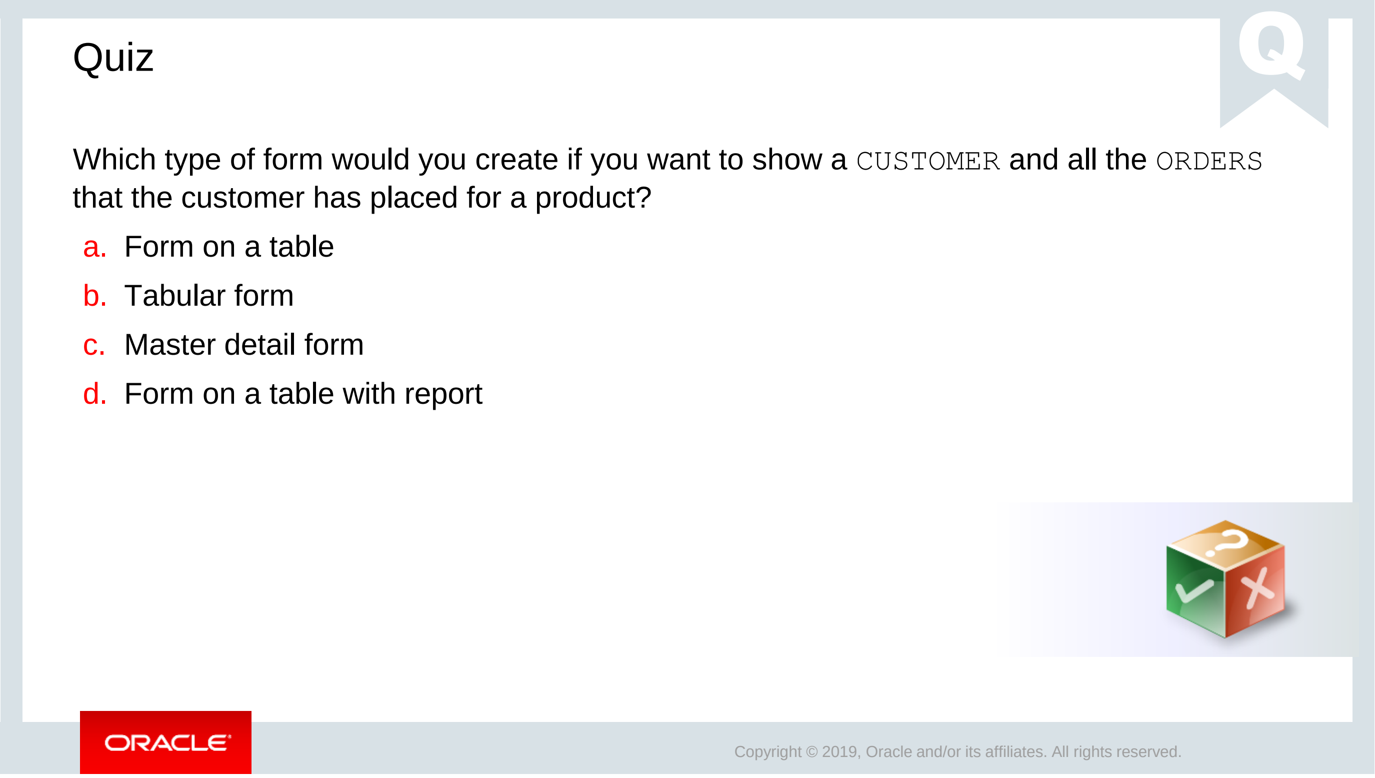
Project Details

□	≡	Project ...	Project ...	Project ...	Project ...	Project ...	Project ...	Project ...	Project ...	Project ...	Project ...	Project ...	Project ...	Project L...
☑	≡	APEX4.2 ...	Developi...	104	15-DEC-14	20-DEC-14	01-APR-15	24-MAR-...	N	-	504	20-DEC-14	504	23-MAR-1...
□	≡	APEX5.0 ...	Developi...	102	01-JAN-15	15-JAN-15	15-APR-15	-	Y	607	504	15-JAN-15	504	20-MAR-1...

A stacked master detail features two editable interactive grids based on two related tables or views. You can select a row in the master grid to update the detail grids. In the example in the slide, the reports page is created, and the master and details information is shown on the same page.

The Form created on the master table will allow users to create a new project type altogether. You can see a report of all *Project Types* when you select the check box beside each row. For example, in the slide above, when you select a particular *Project Type Name*, for example, *Curriculum*, the table below displays the list of all curriculum projects.

Note that if you create/add a new *Project Type* using the **Add Row** option, you can see the newly created Project row in the *Project Details* Report.



Quiz

Which type of form would you create if you want to show a `CUSTOMER` and all the `ORDERS` that the customer has placed for a product?

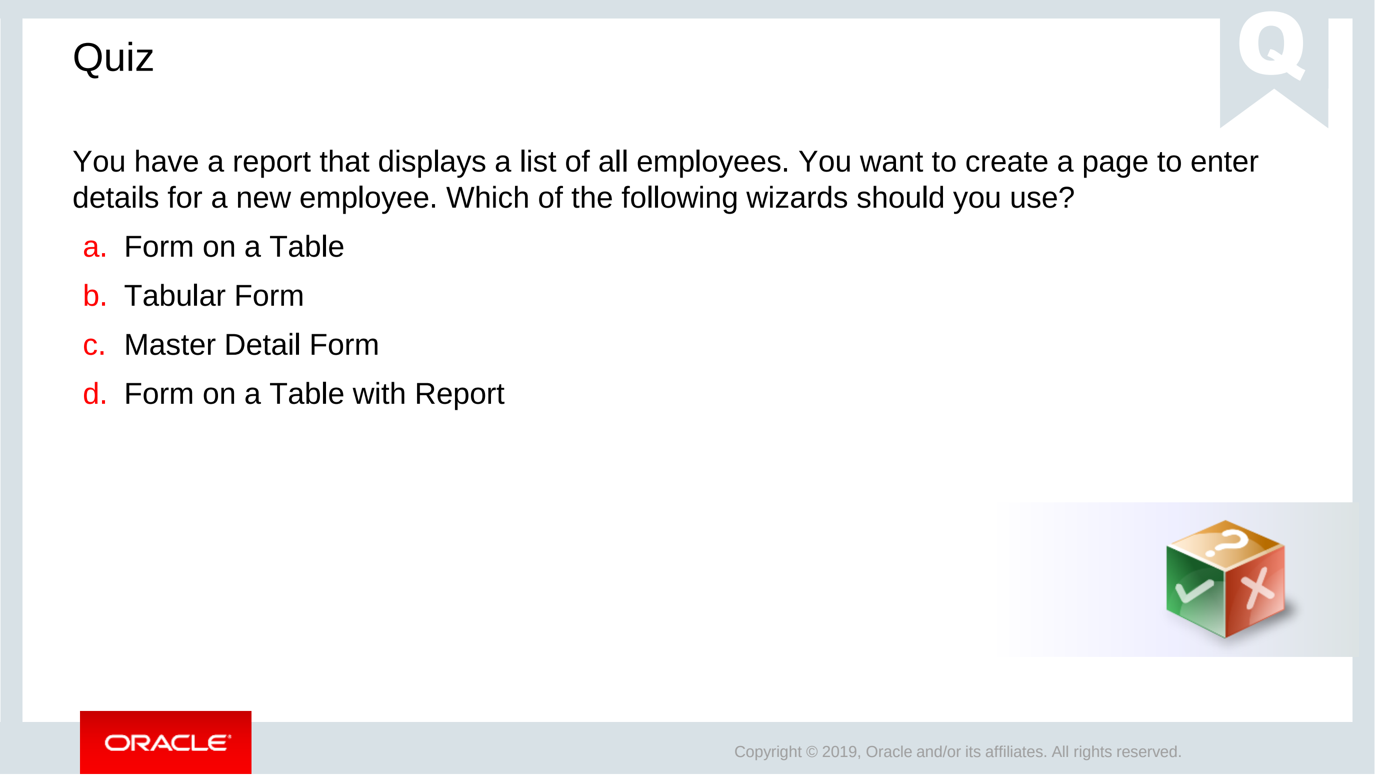
a. Form on a table

b. Tabular form

c. Master detail form

d. Form on a table with report

Answer: c



Quiz

You have a report that displays a list of all employees. You want to create a page to enter details for a new employee. Which of the following wizards should you use?

a. Form on a Table

b. Tabular Form

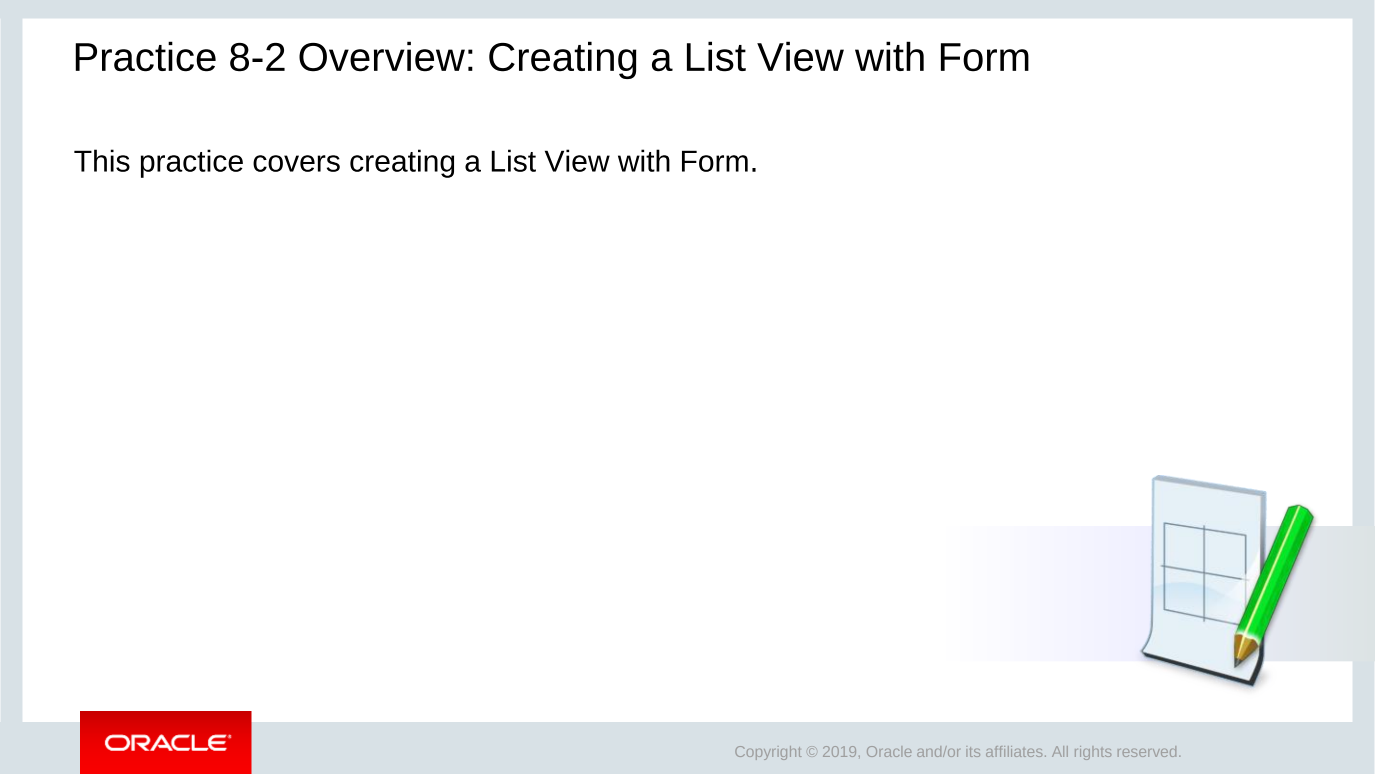
c. Master Detail Form

d. Form on a Table with Report

Answer: a

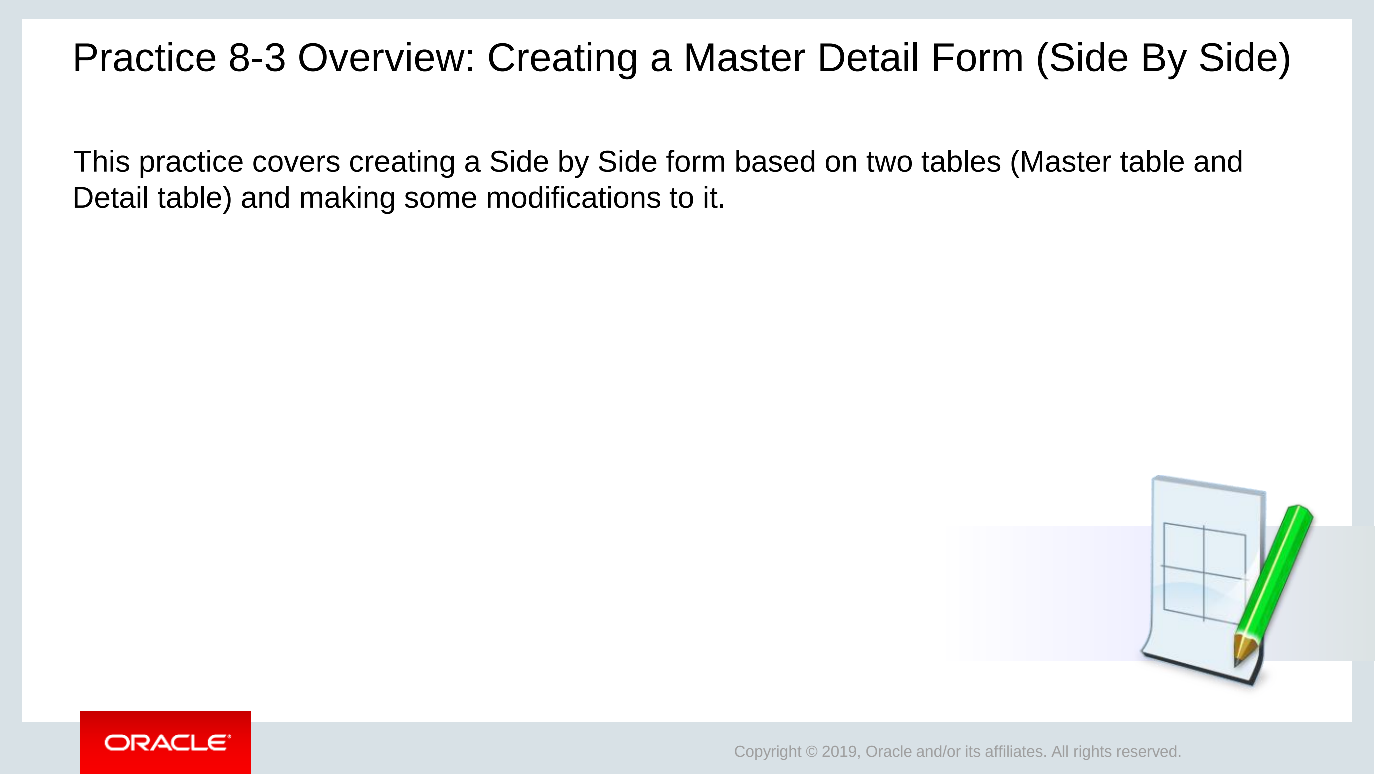
Practice 8-1 Overview: Creating a Report with Form

This practice covers creating a Report with Form on a Table in the GMT application.



Practice 8-2 Overview: Creating a List View with Form

This practice covers creating a List View with Form.



Practice 8-3 Overview: Creating a Master Detail Form (Side By Side)

This practice covers creating a Side by Side form based on two tables (Master table and Detail table) and making some modifications to it.

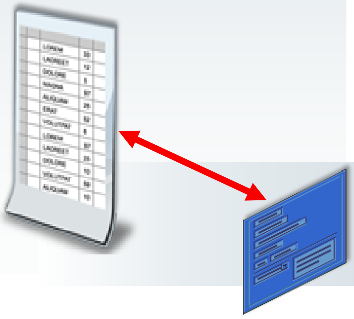
Lesson Agenda

- Introducing Forms
- Creating Forms
- Modifying Forms
  - Linking a Report to a Form
  - Using the Quick Edits Option
  - Reordering Items
  - Changing the Item Display Type
  - Customizing Forms

ORACLE

Linking a Report to a Form

1. From the page definition of the page where you created the report, select Attributes.
2. Locate Link under Properties in the right pane and select “Link to custom target” for Link Column.
3. Click Target and select the target page.
4. Select the item and value that is to be passed with the link.
5. Click the “Save and Run” icon to view the report.

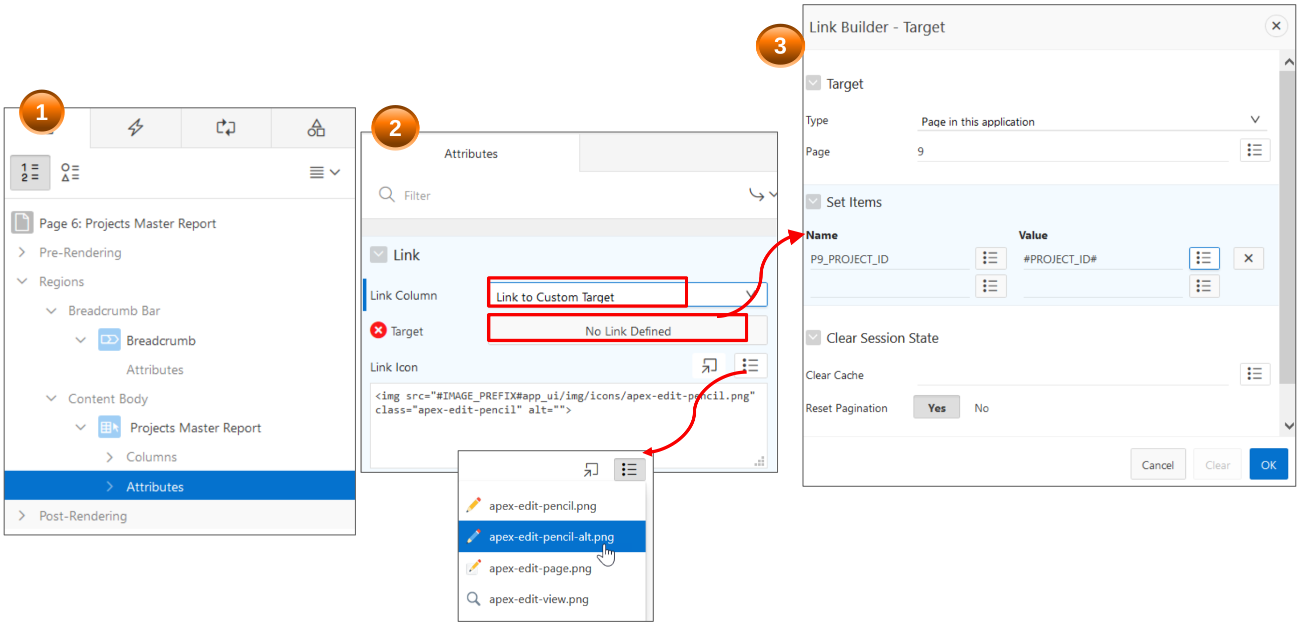


ORACLE

Now that you have created multiple forms, let’s learn how we can link them to an existing report.

In one of the earlier sections, you learned how to create a *Report with a Form on Table*. The *Report with a Form on Table* Wizard automatically creates the required report and form and links them. However, stand-alone reports (interactive or classic) can also be linked to existing forms. This slide provides an overview of the steps to link a report to a form.

Example: Linking a Report to a Form



Link Column: Link to Custom Target

Target: No Link Defined

Link Icon:

```
<img src="#IMAGE_PREFIX#app_ui/img/icons/apex-edit-pencil.png"
class="apex-edit-pencil" alt="">
```

Type: Page in this application

Page: 9

Name	Value
P9_PROJECT_ID	#PROJECT_ID#

Clear Cache:

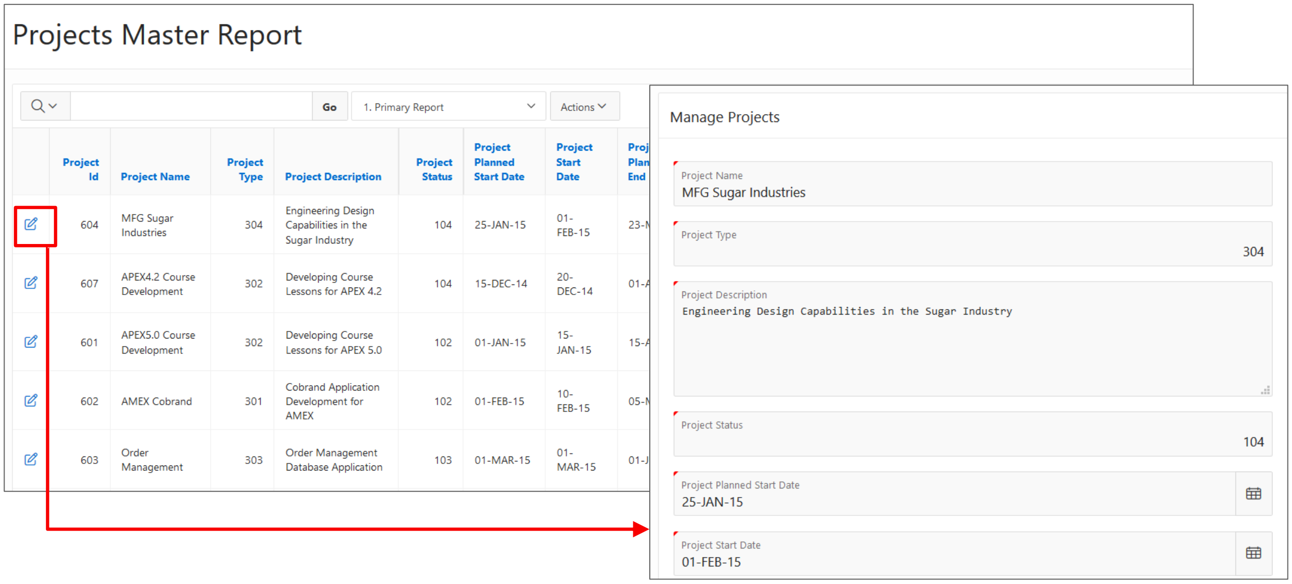
Reset Pagination: Yes No

As you know, Steve had already created a report (*Projects Master Report*) and a form (*Manage Projects*) to manage projects separately. Now, to establish a link between this report and form, Steve updates the report attributes as shown in the slide.

This slide shows how you can link an interactive report to a form.

1. From the page definition of the page where you have created the report, click **Attributes** under Rendering in the left pane.
2. Locate **Link** under Properties in the right pane and select **Link to custom target** for Link Column.
3. Click **Target** and a pop-up window opens. Click the three dots in the Page field and select the target page from All Pages. In this example, the *Manage Projects* form is selected as the target page.
4. Select the item and value that is to be passed with the link. In this example, select `P9_PROJECT_ID` for Name and `#PROJECT_ID#` for *Value* and click **OK**.
5. Click the **Save and Run** icon to view the report. When you click any row in the report, the corresponding form opens. This shows that the *Manage Projects* form is now linked to the *Projects Master Report*.
6. Click the Edit icon beside any row that you want to modify. Update the values and click **Apply changes**. You can see that the changes are reflected in the report for that record.

Example: Linking a Report to a Form



Projects Master Report

	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Proj Plan End
✎	604	MFG Sugar Industries	304	Engineering Design Capabilities in the Sugar Industry	104	25-JAN-15	01-FEB-15	23-M
✎	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14	01-A
✎	601	APEX5.0 Course Development	302	Developing Course Lessons for APEX 5.0	102	01-JAN-15	15-JAN-15	15-A
✎	602	AMEX Cobrand	301	Cobrand Application Development for AMEX	102	01-FEB-15	10-FEB-15	05-M
✎	603	Order Management	303	Order Management Database Application	103	01-MAR-15	01-MAR-15	01-J

Manage Projects

Project Name: MFG Sugar Industries

Project Type: 304

Project Description: Engineering Design Capabilities in the Sugar Industry

Project Status: 104

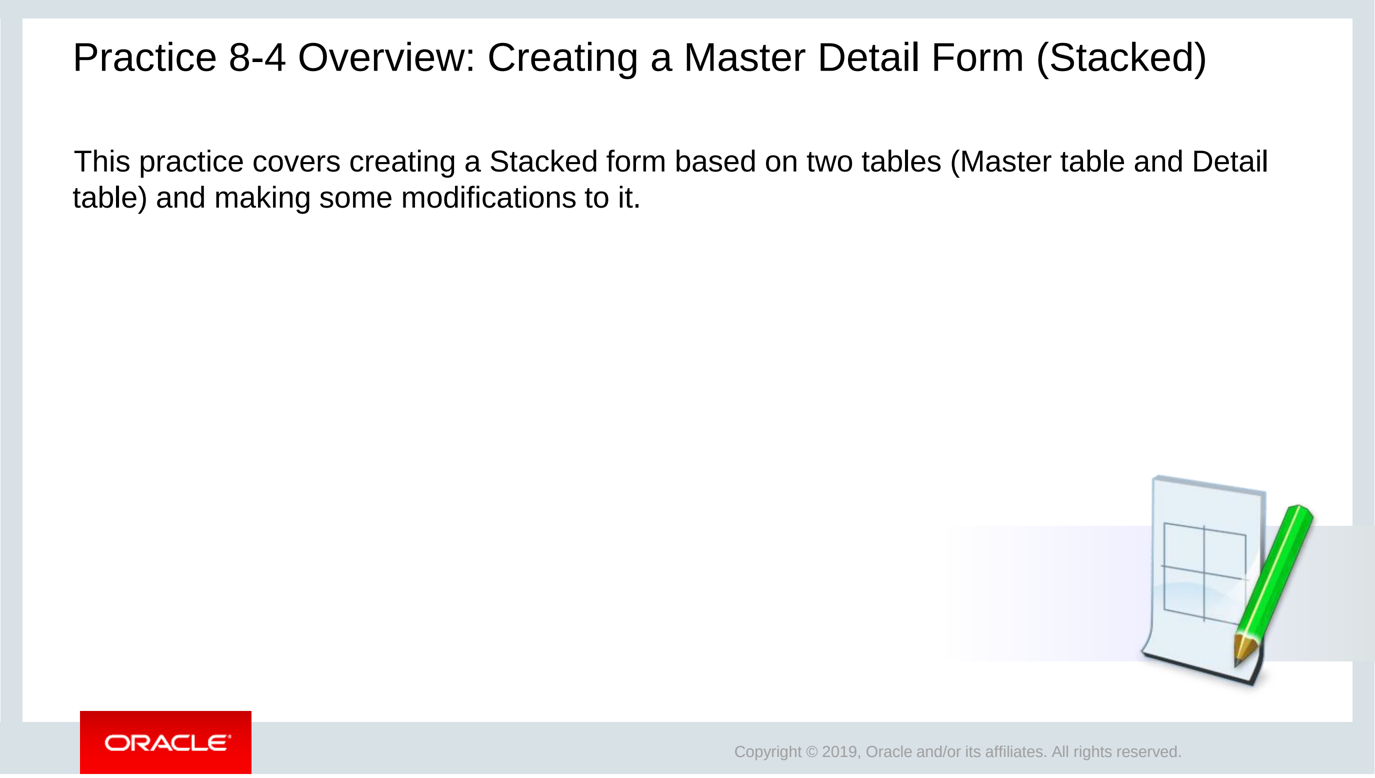
Project Planned Start Date: 25-JAN-15

Project Start Date: 01-FEB-15

As you know, Steve had already created a report (*Projects Master Report*) and a form (*Manage Projects*) to manage projects separately. Now, to establish a link between this report and form, Steve updates the report attributes as shown in the slide.

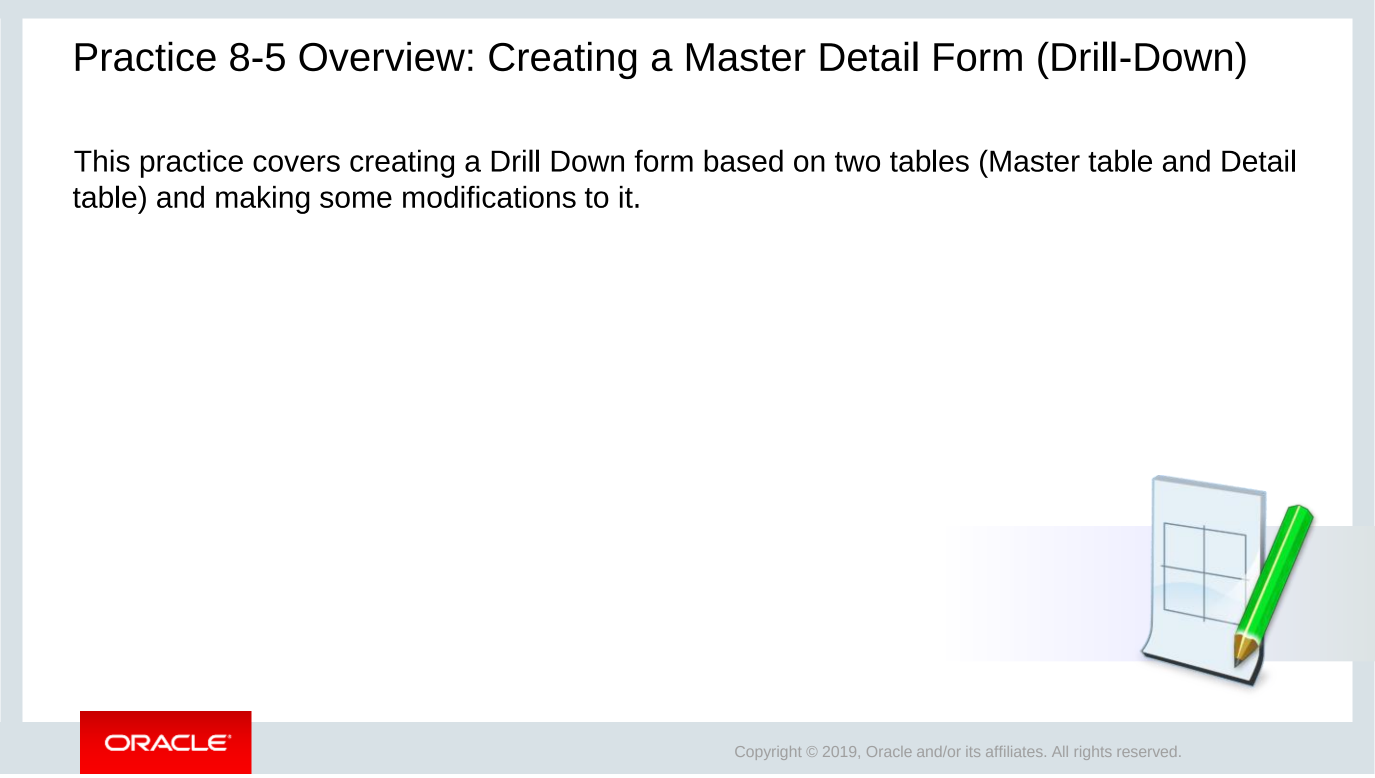
This slide shows how you can link an interactive report to a form.

1. From the page definition of the page where you have created the report, click **Attributes** under Rendering in the left pane.
2. Locate **Link** under Properties in the right pane and select **Link to custom target** for Link Column.
3. Click **Target** and a pop-up window opens. Click the *Up Arrow* beside Page and select the target page from All Pages. In this example, the *Manage Projects* form is selected as the target page.
4. Select the item and value that is to be passed with the link. In this example, select `P9_PROJECT_ID` for Name and `#PROJECT_ID#` for *Value* and click **OK**.
5. Click the **Save and Run** icon to view the report. When you click any row in the report, the corresponding form opens. This shows that the *Manage Projects* form is now linked to the *Projects Master Report*.
6. Click the Edit icon beside any row that you want to modify. Update the values and click **Apply changes**. You can see that the changes are reflected in the report for that record.



Practice 8-4 Overview: Creating a Master Detail Form (Stacked)

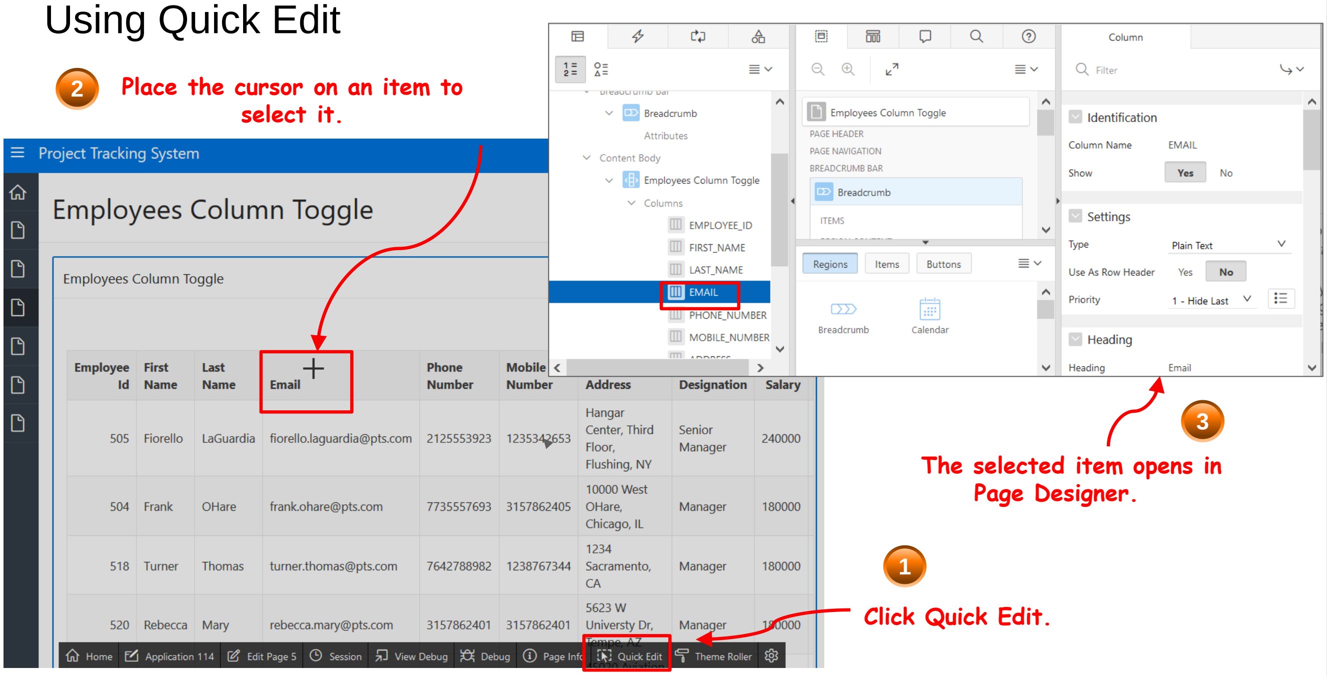
This practice covers creating a Stacked form based on two tables (Master table and Detail table) and making some modifications to it.



Practice 8-5 Overview: Creating a Master Detail Form (Drill-Down)

This practice covers creating a Drill Down form based on two tables (Master table and Detail table) and making some modifications to it.

Using Quick Edit



2 Place the cursor on an item to select it.

3 The selected item opens in Page Designer.

1 Click Quick Edit.

Project Tracking System

Employees Column Toggle

Employees Column Toggle

Employee Id	First Name	Last Name	Email	Phone Number	Mobile Number	Address	Designation	Salary
505	Fiorello	LaGuardia	fiorello.laguardia@pts.com	2125553923	1235342653	Hangar Center, Third Floor, Flushing, NY	Senior Manager	240000
504	Frank	OHare	frank.ohare@pts.com	7735557693	3157862405	10000 West OHare, Chicago, IL	Manager	180000
518	Turner	Thomas	turner.thomas@pts.com	7642788982	1238767344	1234 Sacramento, CA	Manager	180000
520	Rebecca	Mary	rebecca.mary@pts.com	3157862401	3157862401	5623 W University Dr, Tempe, AZ	Manager	180000

Identification

Column Name: EMAIL

Show: Yes [✓ Yes] No

Settings

Type: Plain Text

Use As Row Header: Yes [✓ No]

Priority: 1 - Hide Last

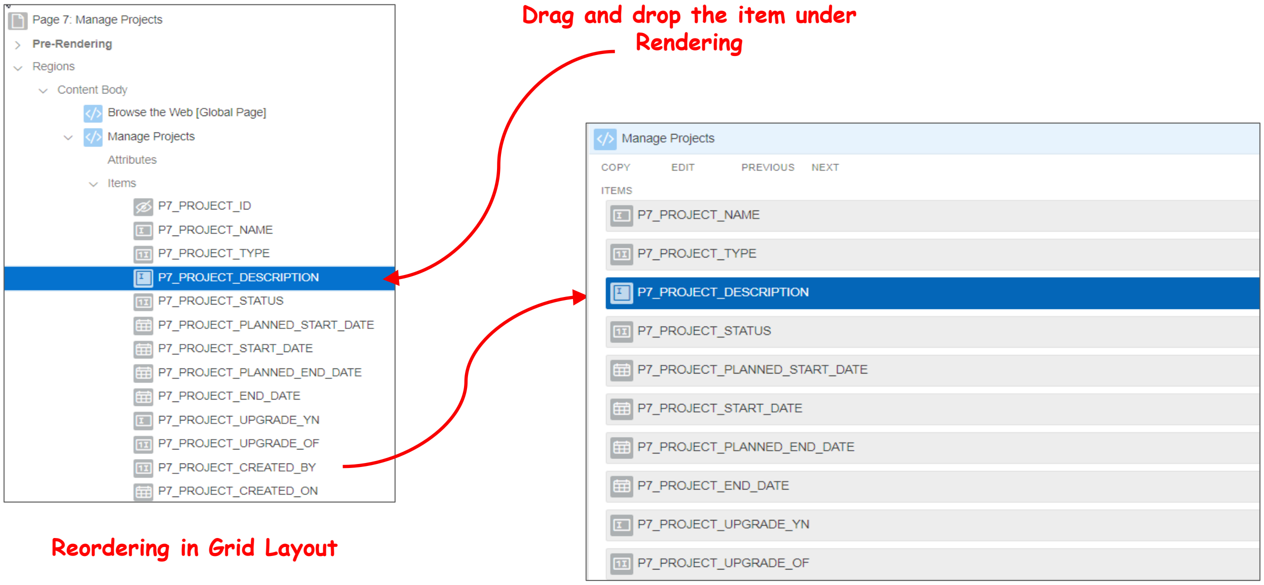
Heading

Heading: Email

The slide shows the *Employees Column Toggle* report that you created in one of the earlier lessons. Steve shows you how you can quickly edit a particular column in this report. In this slide, for example, he suggests using the **Quick Edit** option on the *Runtime Developer Toolbar* to access the *PTS* application, if there is a need to update the salary column.

When you run your form page, you can modify its objects by using the **Quick Edit** button on the Developer toolbar. When you click **Quick Edit**, the cursor changes to a **+** symbol and highlights the item wherever the cursor is placed in the form. Click the highlighted icon to open the page in Page Designer with the same item preselected. Because the page opens with the item selected already, you can see its properties right away in the property editor and make the required modifications quickly. This is useful, for example, when changing a label or the format of an item.

Reordering Items



Drag and drop the item under Rendering

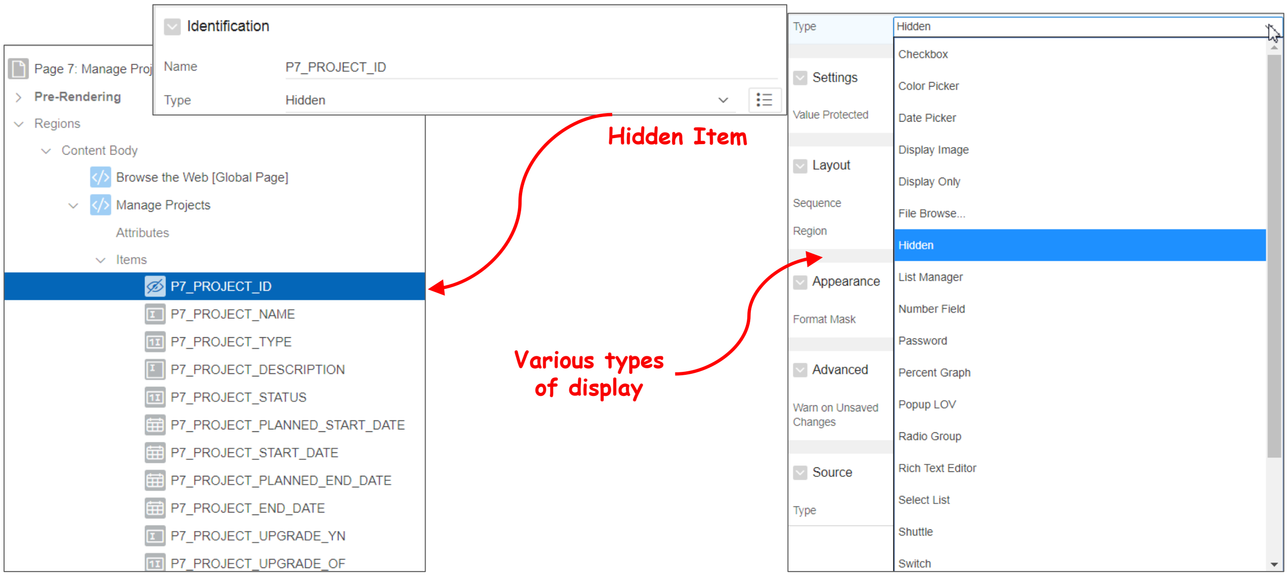
Reordering in Grid Layout

Let’s also see some more user-friendly options provided by Oracle Application Express, such as reordering items, changing display types, and so on.

To change the order in which items appear in your application, you can reorder your items. For example, the `P7_PROJECT_DESCRIPTION` item can be re-ordered to appear before the `P7_PROJECT_TYPE` item. The items on a form can be reordered in Page Designer in two ways:

- Drag the item to the desired position among other items under the Rendering tab.
- Drag the item to the desired position in the Layout of Page Designer.

Changing Item Display Type



Identification

Name: P7_PROJECT_ID

Type: Hidden

Type: Hidden

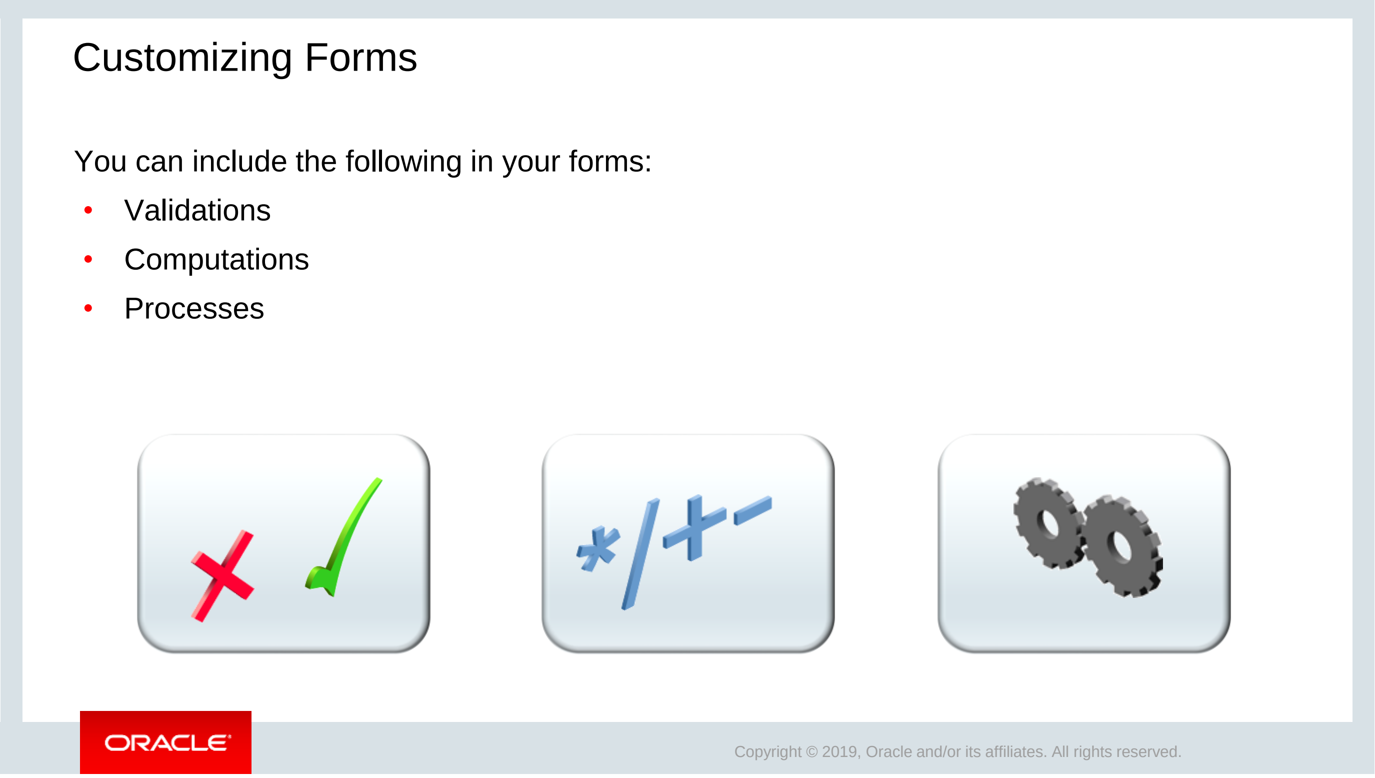
Hidden Item

Various types of display

By default, your varchar/number type columns are displayed as text fields in a form. You can change this default type to other available types, such as drop-down lists, option buttons, check boxes, and pop-up LOVs.

To change the display type for an item, perform the following steps:

1. Select the item on the **Rendering** tab.
2. In the **Property Editor**, select the new type from the **Type** drop-down list.
3. You can modify the display type as a select list, Popup LOV, Date Picker, Color Picker, and so on based on the requirement.

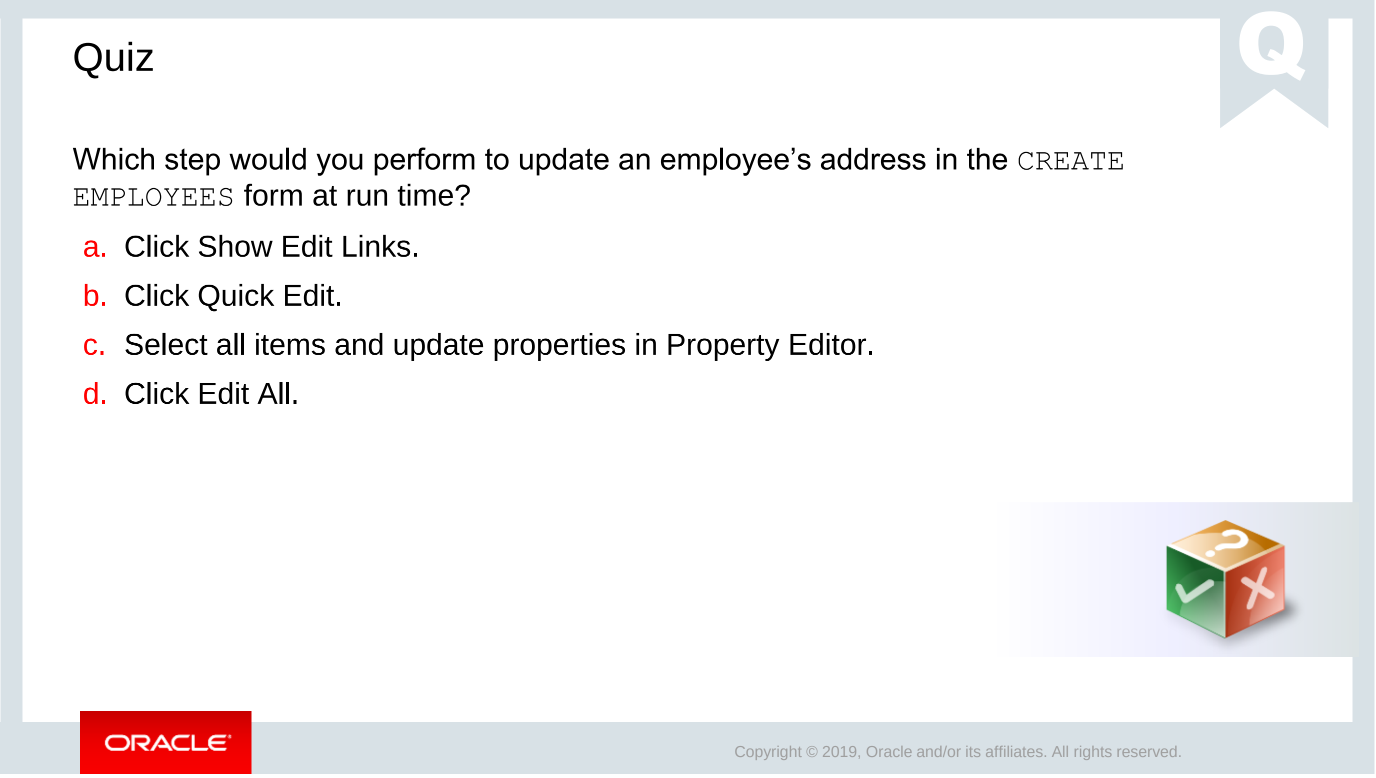


Customizing Forms

You can include the following in your forms:

- Validations
- Computations
- Processes

You can also customize your forms by creating computations and processes. You can also include validations to verify user inputs. You learn about these topics in detail in the lesson titled “Adding Page Processing.”



Quiz

Which step would you perform to update an employee's address in the `CREATE EMPLOYEES` form at run time?

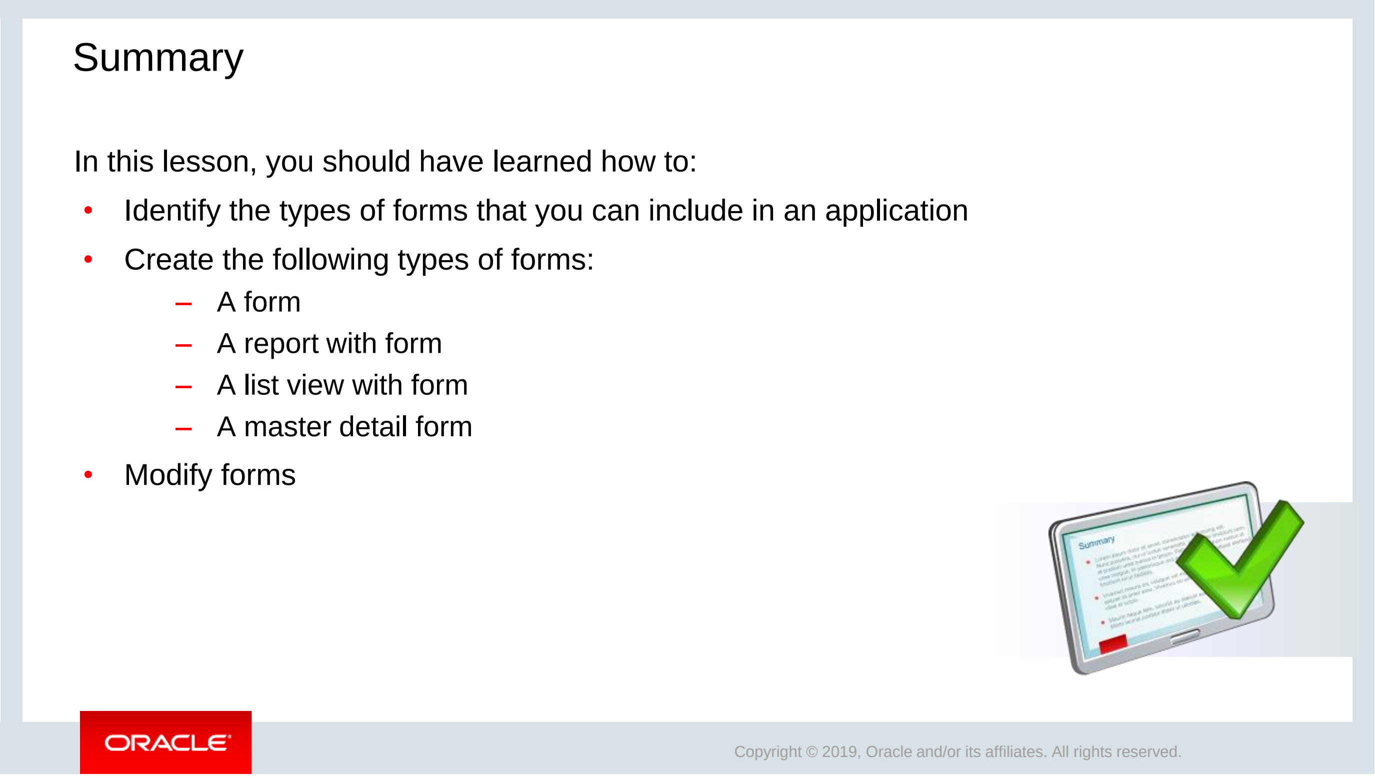
a. Click Show Edit Links.

b. Click Quick Edit.

c. Select all items and update properties in Property Editor.

d. Click Edit All.

Answer: b

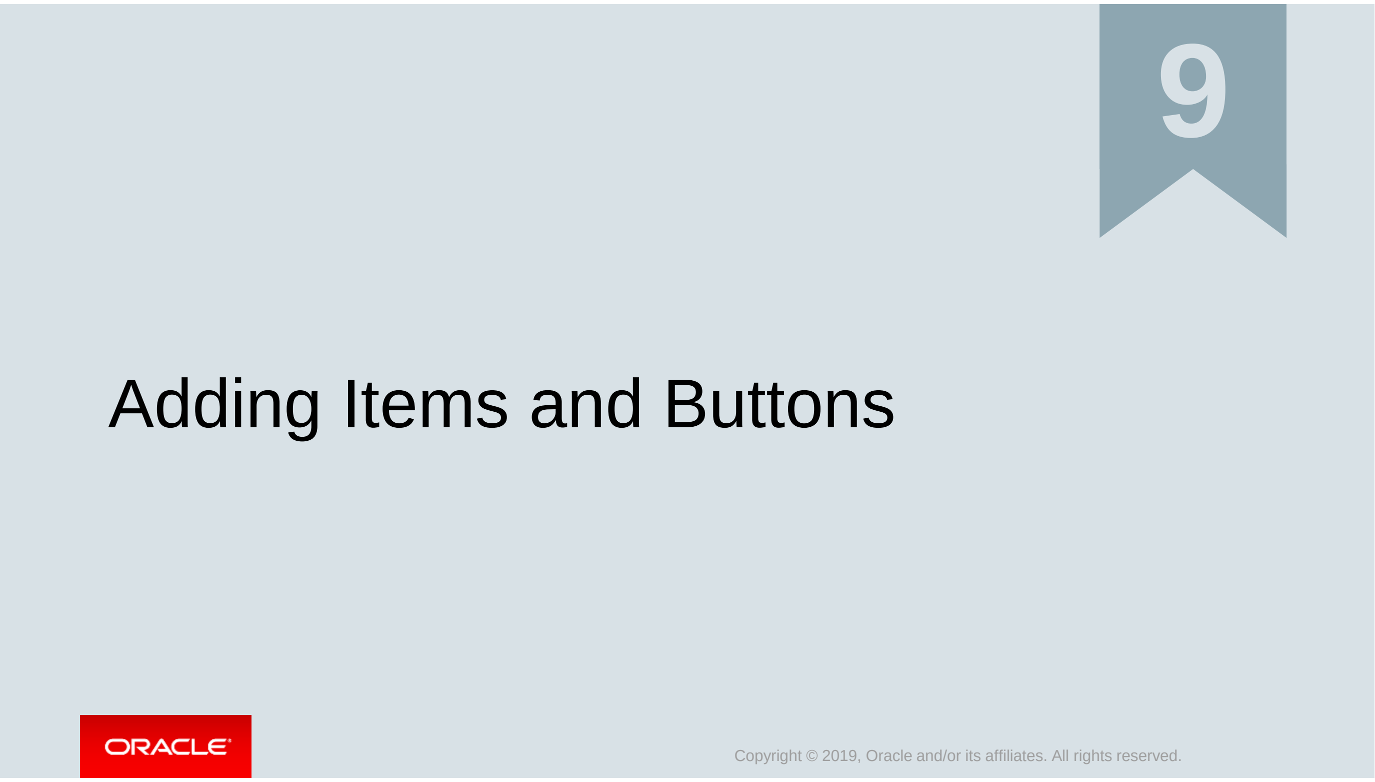


Summary

In this lesson, you should have learned how to:

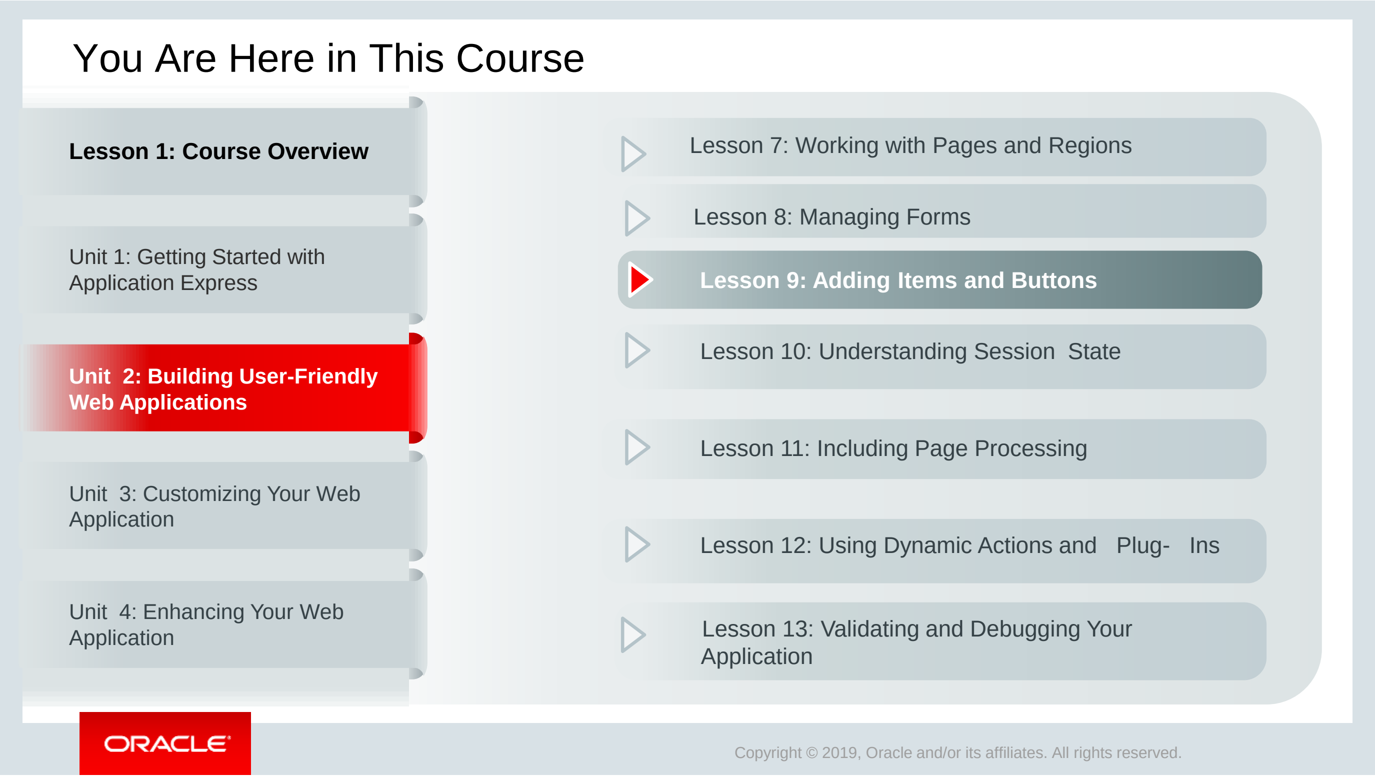
- Identify the types of forms that you can include in an application
- Create the following types of forms:
  - A form
  - A report with form
  - A list view with form
  - A master detail form
- Modify forms

This lesson showed you how to create forms, how to use the various built-in wizards that help you create forms, and how to edit the attributes of a form.



9

Adding Items and Buttons



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

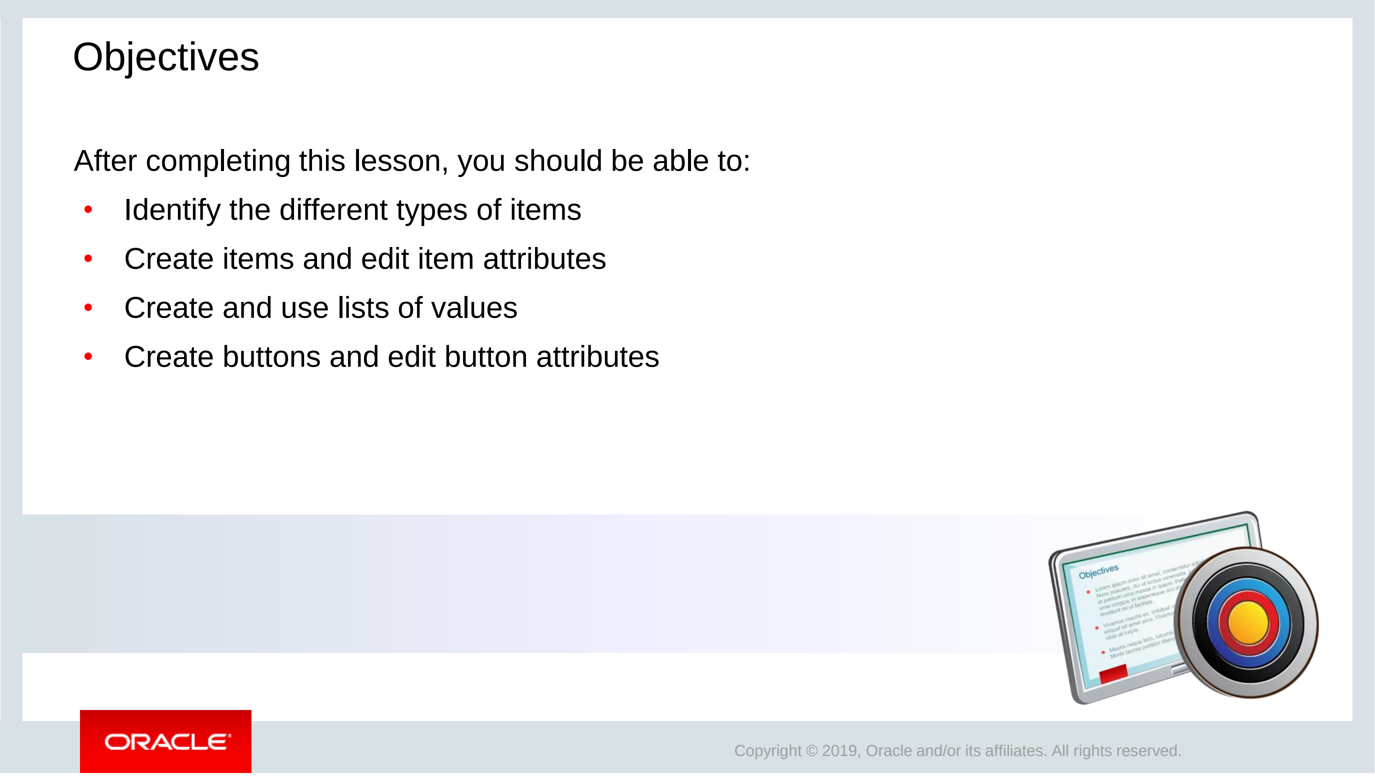
Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- Lesson 7: Working with Pages and Regions
- Lesson 8: Managing Forms
- **Lesson 9: Adding Items and Buttons**
- Lesson 10: Understanding Session State
- Lesson 11: Including Page Processing
- Lesson 12: Using Dynamic Actions and Plug-Ins
- Lesson 13: Validating and Debugging Your Application

This slide is a graphical depiction of the course, particularly highlighting Unit 2 - Lesson 9, which is dealt with in these slides.



Objectives

After completing this lesson, you should be able to:

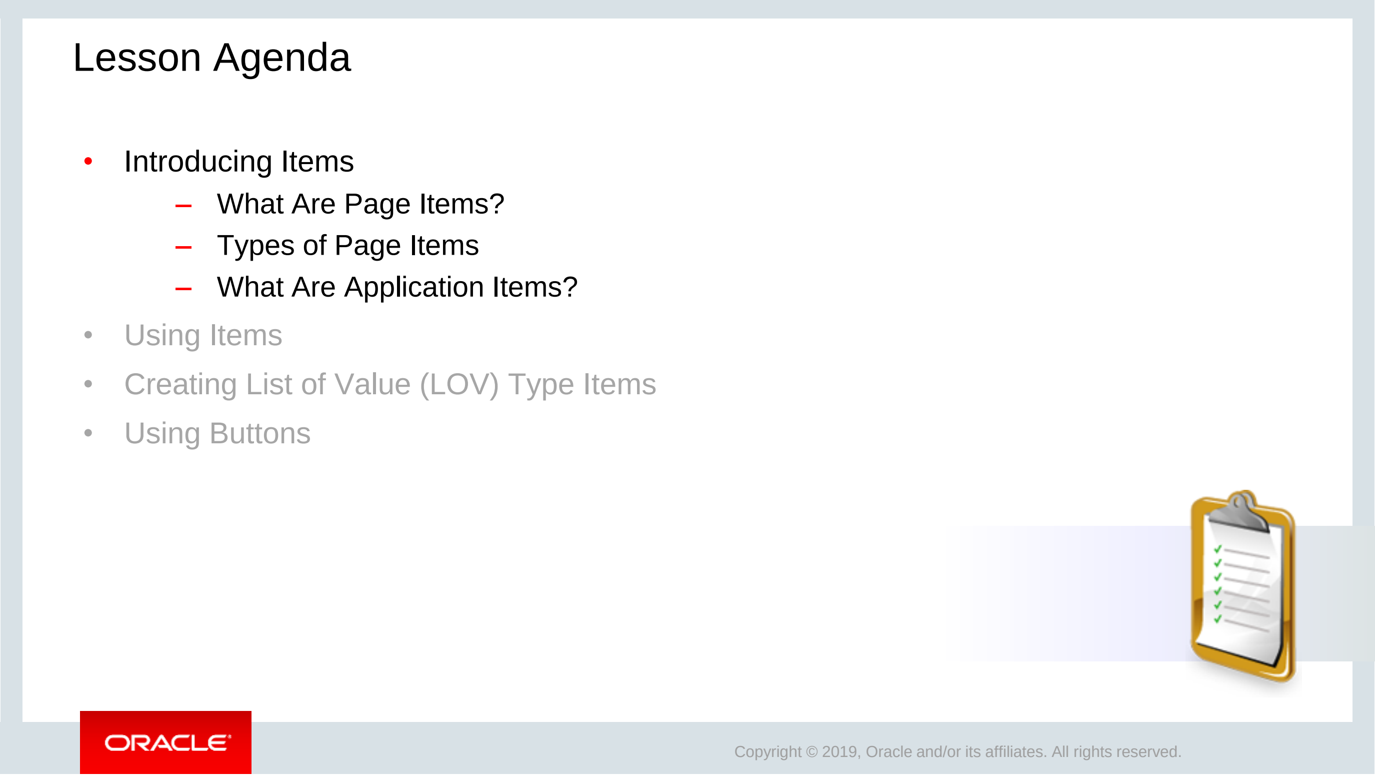
- Identify the different types of items
- Create items and edit item attributes
- Create and use lists of values
- Create buttons and edit button attributes

In this lesson, you learn how to include items and buttons on application pages. You also learn how to create a list of values (which is a shared component) and associate it with the supported item types.

Steve Enhances User Interface—Adds Items and Buttons to Pages

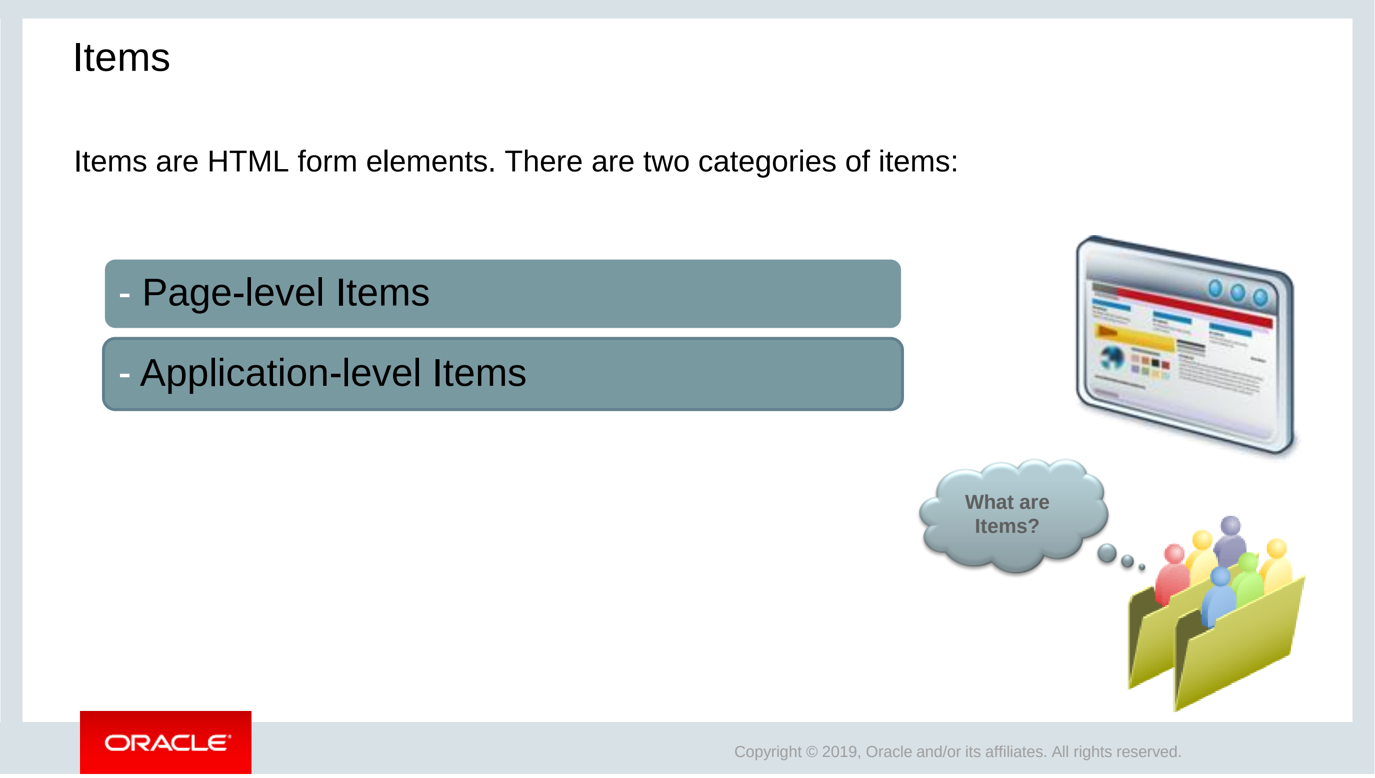
By this time, Steve has created multiple pages as he was building the Reports and Forms for the *Project Tracking System (PTS)* application. However, the pages that he created are not functional. To make the pages functional, he plans to add items and buttons to the page (you will learn about items and buttons in the following slides).

He decides to look up the various items and buttons available in Oracle Application Express and incorporate them in the *PTS* application accordingly. In this lesson, as he tries out these features of Oracle Application Express, he switches at times between the *Sample Database Application* and his *Project Tracking System* application.



Lesson Agenda

- Introducing Items
  - What Are Page Items?
  - Types of Page Items
  - What Are Application Items?
- Using Items
- Creating List of Value (LOV) Type Items
- Using Buttons



Items

Items are HTML form elements. There are two categories of items:

- Page-level Items
- Application-level Items

You learned about pages in an earlier chapter. Now, let’s see what are items.

Items are HTML form elements such as text fields, select lists, and check boxes with an associated session item so that it can be retrieved at a later time. Item attributes affect the display and behavior of items on a page. For example, these attributes can impact where a label is displayed, how large an item is, and whether the item is displayed next to or below the previous item.

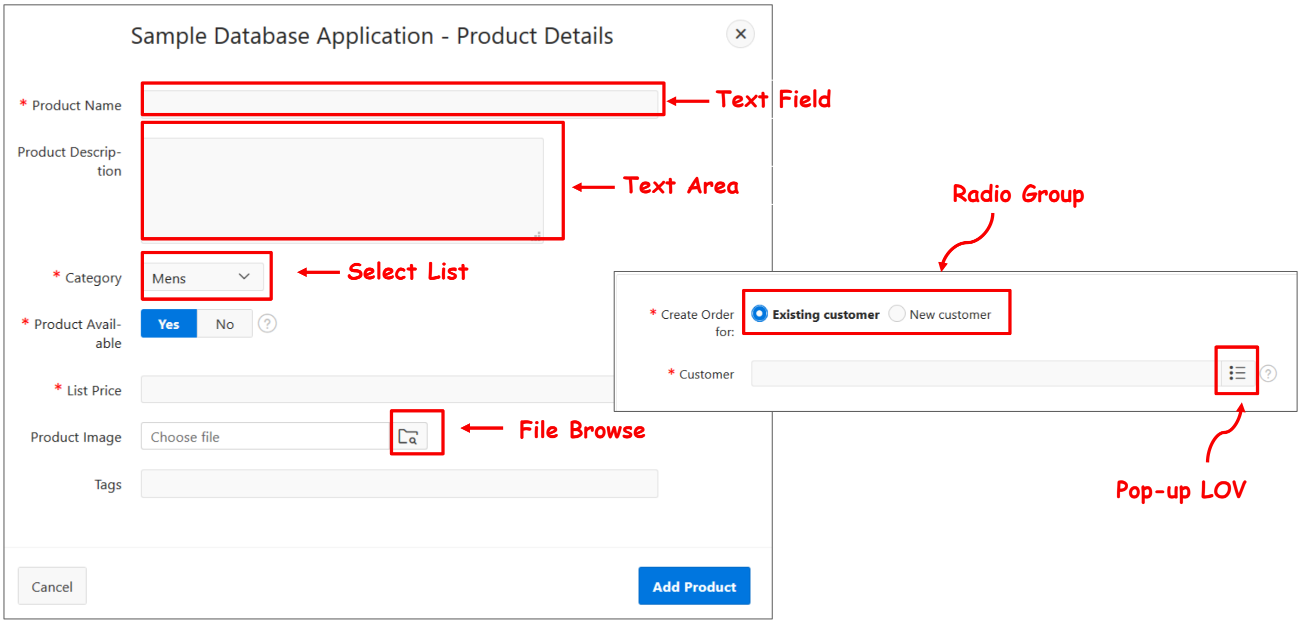
There are two types of items:

- *Page Items*: Page-level items are placed on a page and have associated user interface properties.
- *Application Items*: Application-level items are not associated with a page and, therefore, have no user interface properties. You can use an application item as a global variable.

We will again talk about these two item types in a later lesson on *Page Processing* when we are discussing page-level and application-level computations.

In the next two slides, you get a graphical representation of what each type of item is.

What Are Page Items?



Text Field

Text Area

Select List

File Browse

Radio Group

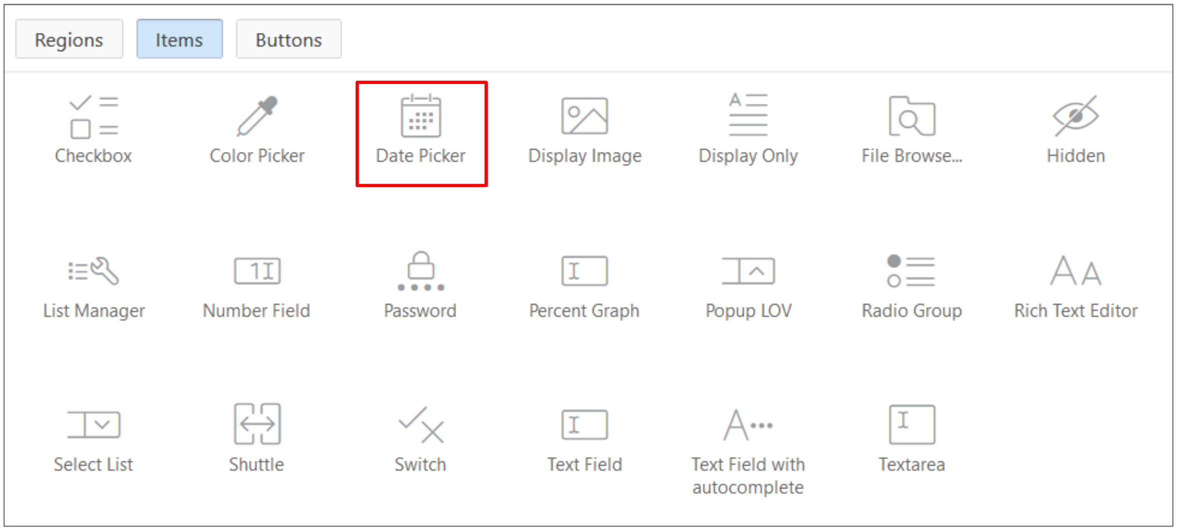
Pop-up LOV

Look at this screenshot from the *Products* and *Orders* pages of the *Sample Database Application*. Can you see the different page items here?

Page items are placed on a page and have associated user interface properties, such as Display As, Label, and Label Templates. Another type of item, application items, is discussed in a later slide.

When you create a form by using a wizard, an item is created for each column of the table. The default item type is a text field, text area, number field, date picker, or File Browse depending on whether the database table column type is varchar, varchar2 (with size greater than 255 characters), numeric, date, respectively. You can edit the item properties to change the display type. For example, you can change a text field to a text area or select list.

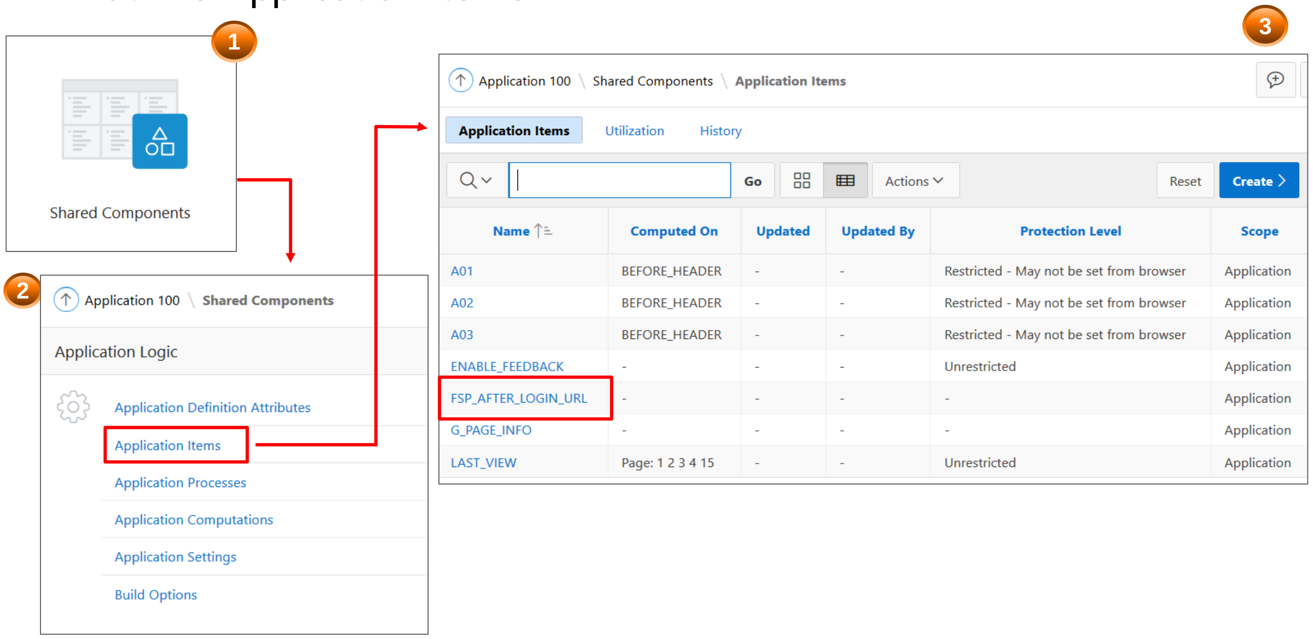
Types of Page Items



You learned about the *Items* and *Buttons* tabs in a previous lesson on the Page Designer UI elements. Just to recollect, the *Regions*, *Items*, and *Buttons* tabs appear in the Page Designer (Central Pane: *Gallery*). Oracle Application Express provides multiple item types. In a later slide, you will be creating a *Date Picker* item type.

You can learn more about the behavior of item type by reading Oracle Application Express documentation (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/about-item-types.html#GUID-81BF829A-B26C-484D-A321-6A3F47C9192B).

What Are Application Items?



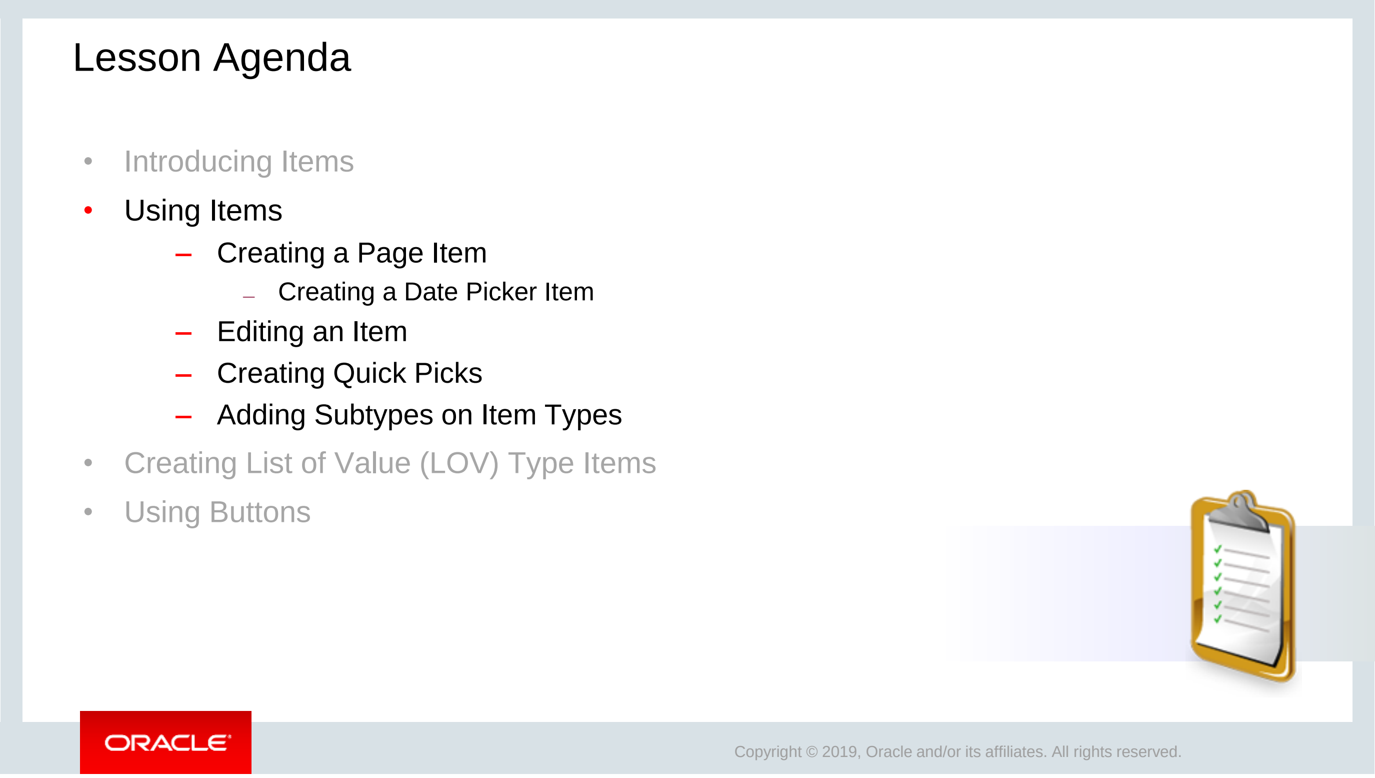
Name	Computed On	Updated	Updated By	Protection Level	Scope
A01	BEFORE_HEADER	-	-	Restricted - May not be set from browser	Application
A02	BEFORE_HEADER	-	-	Restricted - May not be set from browser	Application
A03	BEFORE_HEADER	-	-	Restricted - May not be set from browser	Application
ENABLE_FEEDBACK	-	-	-	Unrestricted	Application
FSP_AFTER_LOGIN_URL	-	-	-	-	Application
G_PAGE_INFO	-	-	-	-	Application
LAST_VIEW	Page: 1 2 3 4 15	-	-	Unrestricted	Application

Now that it is clear about page items, let's see what are application items.

Application items are not associated with a page and, therefore, have no user interface properties. An application item can be thought of as a global variable. Application items are typically configured by using processes or computations (you would learn about these in a later chapter) or by passing values in a URL. As you see in the above example, the `FSP_AFTER_LOGIN_URL` application item is used internally by Oracle Application Express to remember the page that users attempted to visit before they were shown the login page.

You create an application item from the *Application Items* page. To access the Application Items page, perform the following steps:

1. Click the **Shared Components** icon on the application home page.
2. In the Logic pane, click **Application Items**.
   The Application Items page is displayed. You can click the item icon to view or edit details.

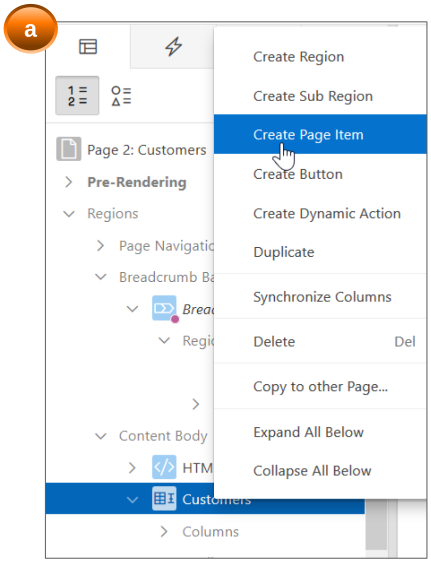


Lesson Agenda

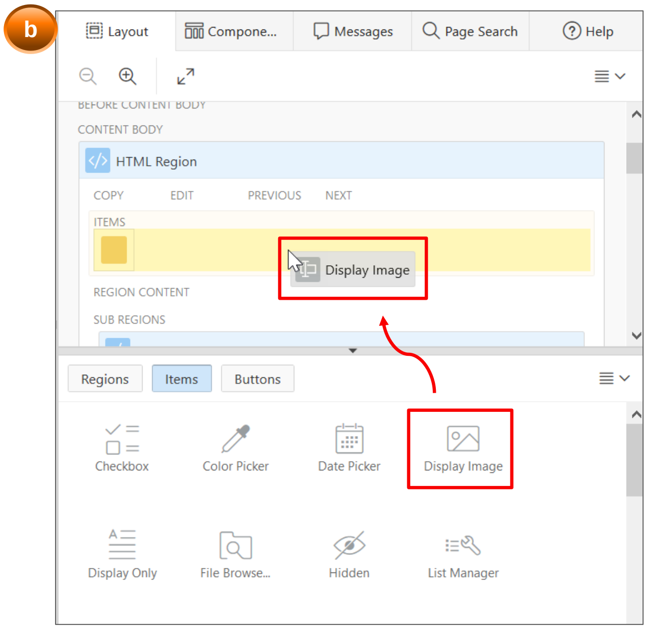
- Introducing Items
- Using Items
  - Creating a Page Item
    - Creating a Date Picker Item
  - Editing an Item
  - Creating Quick Picks
  - Adding Subtypes on Item Types
- Creating List of Value (LOV) Type Items
- Using Buttons

ORACLE

Creating a Page Item



OR



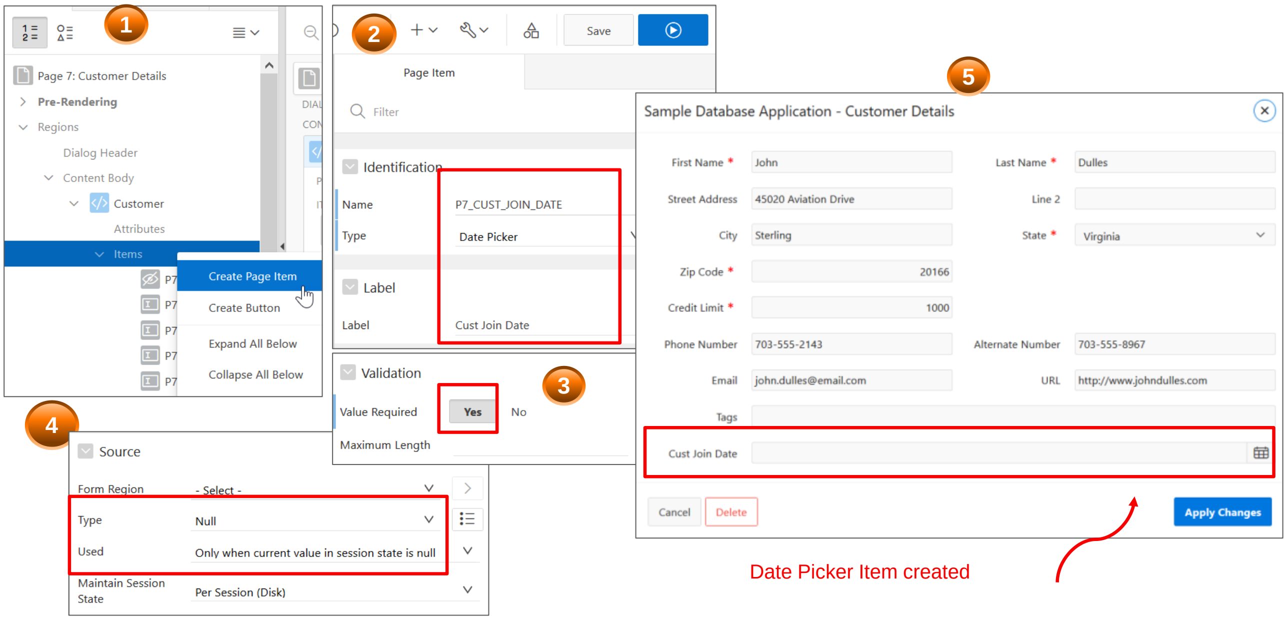
You have learned about page and page item types, but how do you create a page item?

You can create a page item in either of the following ways:

- In the Rendering pane of the Page Designer, right-click the region node where you want to create the item and select **Create Page Item**.
- Choose the item that you want to create from the Items gallery. Drag it from the gallery to the Layout of the page under the region where you want to create the item.

After either of the above steps, set/edit the property of the item by using the Property Editor (right pane of the Page Designer). In this example, we select the *Display Image* as the Page Item type.

Example: Creating a Date Picker Item



Name: P7_CUST_JOIN_DATE
Type: Date Picker
Label: Cust Join Date
Value Required: Yes No
Maximum Length

Form Region: - Select -
Type: Null
Used: Only when current value in session state is null
Maintain Session State: Per Session (Disk)

Sample Database Application - Customer Details

First Name *	John	Last Name *	Dulles
Street Address	45020 Aviation Drive	Line 2	
City	Sterling	State *	Virginia
Zip Code *	20166		
Credit Limit *	1000		
Phone Number	703-555-2143	Alternate Number	703-555-8967
Email	john.dulles@email.com	URL	http://www.johndulles.com
Tags			
Cust Join Date			

Date Picker Item created

Now, Steve wants to know when a new customer was created or the joining date of a new customer. And for that he needs to add a calendar icon next to the *Customer Join Date* field. He was thinking that he can add a similar Date Picker item for the *Create Employees* form in his PTS application too. So, whenever a new employee joins, his/her joining date can be added easily.

Date Picker item displays a text field with a calendar icon. You can either enter the date directly into the text item or click the calendar icon to select a date and, optionally, a time from the calendar pop-up. Time is only displayed in the calendar pop-up if the Format Mask (under *Appearance*) for this item includes time components.

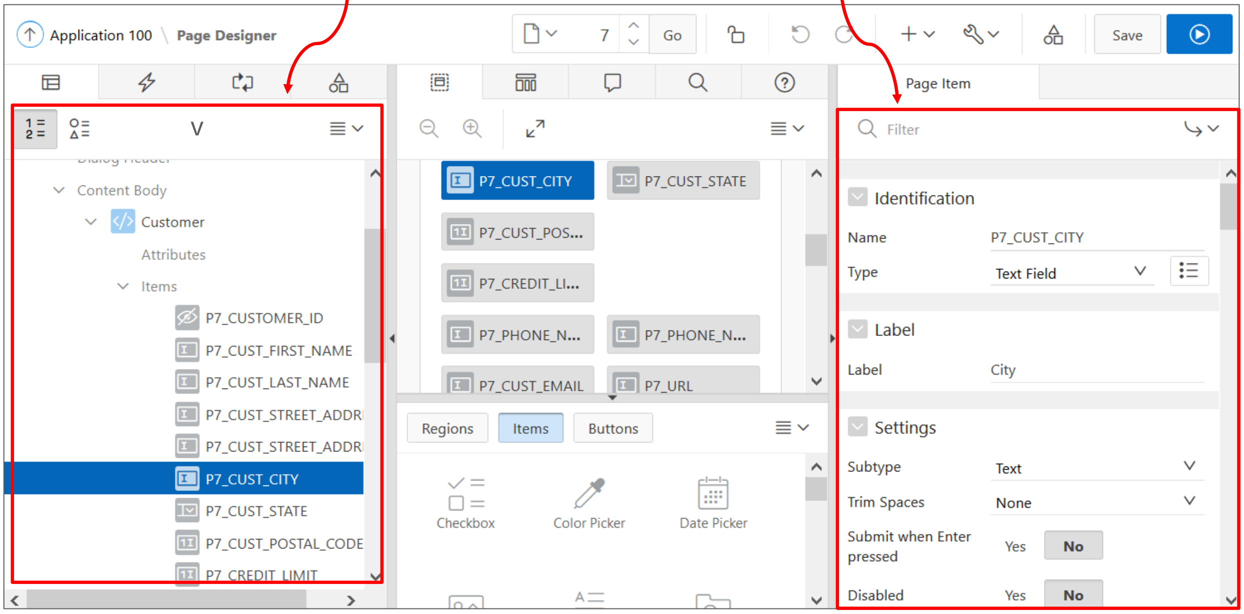
You can create a *Date Picker* item in either of the ways mentioned in the previous slide. After either of the above steps, select the item in the Layout or the Rendering pane and perform the following steps:

1. In the **Identification** section of the Property Editor pane, enter a name for the item. As a best practice, use the format `P<n>_<ITEM_NAME>` to name the item. Select the **Type** as **Date Picker**. Scroll down. **Note:** If you used the drag-and-drop option to create the item, selecting the type will not be necessary.
2. Accept the defaults or change the item Label, Settings, and Layout. Scroll down.
3. You can specify whether a value is required for the item. Under Validation, if you select **Yes** for Value Required, the item is validated to ensure that it is not null when the page is submitted. The options in this step may differ for each item type. For the date picker item, you can specify a format mask, template, and so on.
4. You can apply custom CSS to your item. Specify the **Source** for the item. You can also specify a default value for the item.
5. You can specify a condition for the item so that the item appears only when the condition is true. Also, you can specify whether the item is read-only or not, the various security parameters, the help text, and so on.
6. Save and run the page to check whether the item was created successfully.

Editing an Item

Rendering pane

Property Editor



Identification

Name: P7_CUST_CITY

Type: Text Field

Label

Label: City

Settings

Subtype: Text

Trim Spaces: None

Submit when Enter pressed: Yes [✓ No]

Disabled: Yes [✓ No]

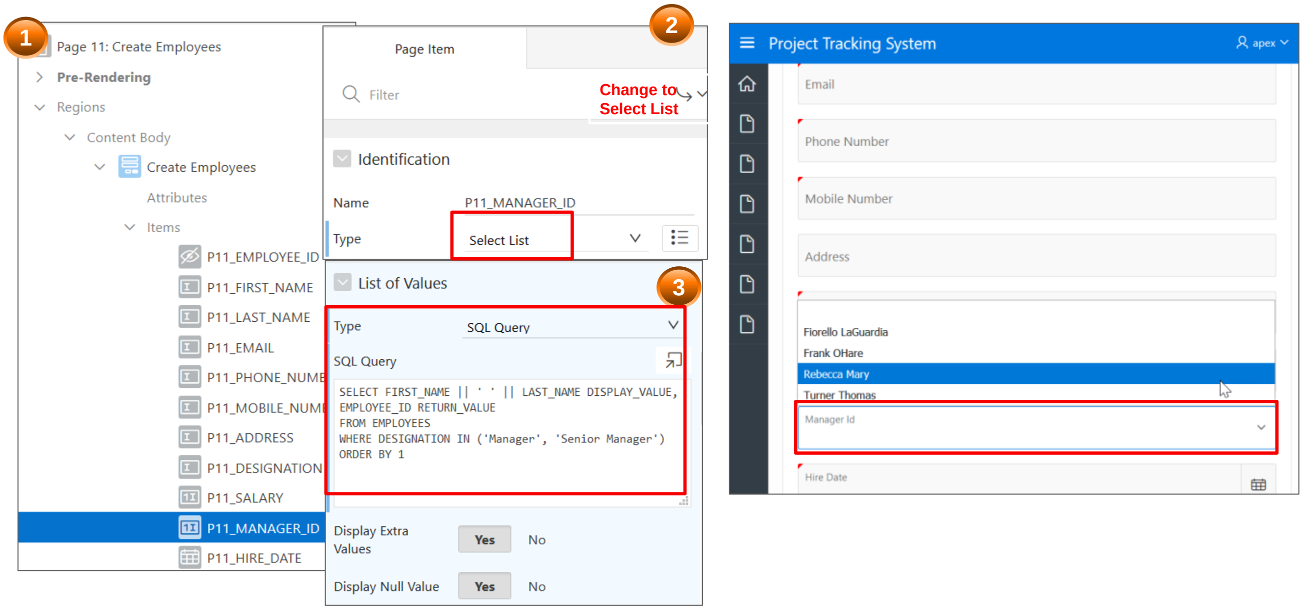
You now know how to create an item. Let's see how you can edit existing items.

To edit an item, navigate to the Page Definition and select the item you want to edit in the Rendering pane of the Page Designer. When you select the item, you notice that the Property Editor on the right pane displays the list of attributes that you can edit for that particular item. These attributes are categorized as follows:

- Identification, Label, Settings, Layout, Appearance, Validation, Advanced, Source, Default, Quick Picks, Server-side Condition, Read Only, Security, Configuration, Help, Comments, Audit Information

In the next slide, you see an example of how you can edit an item.

Examples: Editing an Item



Identification

Name P11_MANAGER_ID

Type Select List

List of Values

Type SQL Query

SQL Query

```
SELECT FIRST_NAME || ' ' || LAST_NAME DISPLAY_VALUE,
EMPLOYEE_ID RETURN_VALUE
FROM EMPLOYEES
WHERE DESIGNATION IN ('Manager', 'Senior Manager')
ORDER BY 1
```

Display Extra Values Yes No

Display Null Value Yes No

Steve now switches to his *PTS* application. He wants to convert *Manager ID* on *Create Employees* Form into a *Select List* from *Number Field.* He would later change it into a *Popup LOV*, to sort the employees by their manager IDs. Let’s see how he does it by editing the *Manager ID* item.

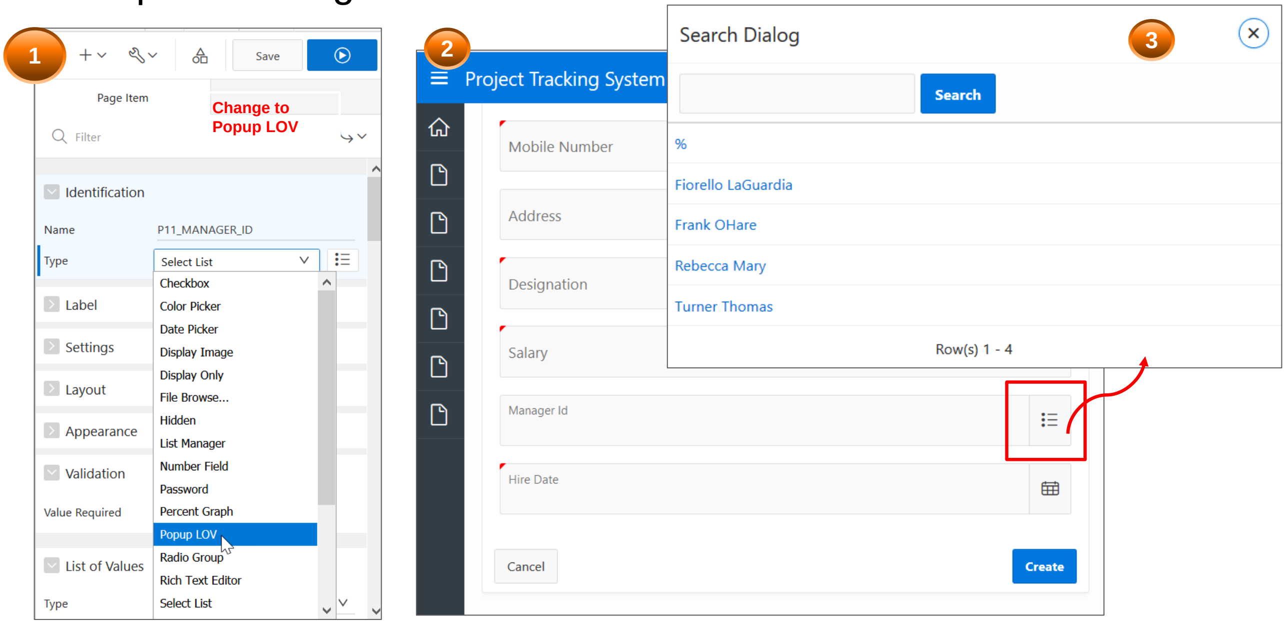
To change *Manager_ID* item from a *Number Field* to a *Select List:*

1. Open *Create Employees* Form in Page Designer view.
2. Under Rendering, click **P11_MANAGER_ID** under Page Items (screenshot 1).
3. Under Identification in Properties Pane on the right side, change Type from *Number Field* to *Select List* (screenshot 1). An error message appears under Messages on the page.
4. To resolve that, locate *List of Values* in the Property Pane. Enter the following values:
   - **List of Values: Type**: Select **SQL Query**
   - **List of Values: SQL Query** (code text area) (screenshot 3): Enter

     ```
     SELECT FIRST_NAME || ' ' || LAST_NAME DISPLAY_VALUE,
     EMPLOYEE_ID RETURN_VALUE
     FROM EMPLOYEES
     WHERE DESIGNATION IN ('Manager', 'Senior Manager')
     ORDER BY 1
     ```
   - **Display Extra Values:** Select **No**
   - **Display Null Value:** Select **No**
5. Click **Save and Run** icon, and you can find all manager names as Select List on the form for *Manager ID* field.

In the next slide, you will change this **Select List** item to a **Popup LOV**.

Examples: Editing an Item



Change to Popup LOV

Name P11_MANAGER_ID

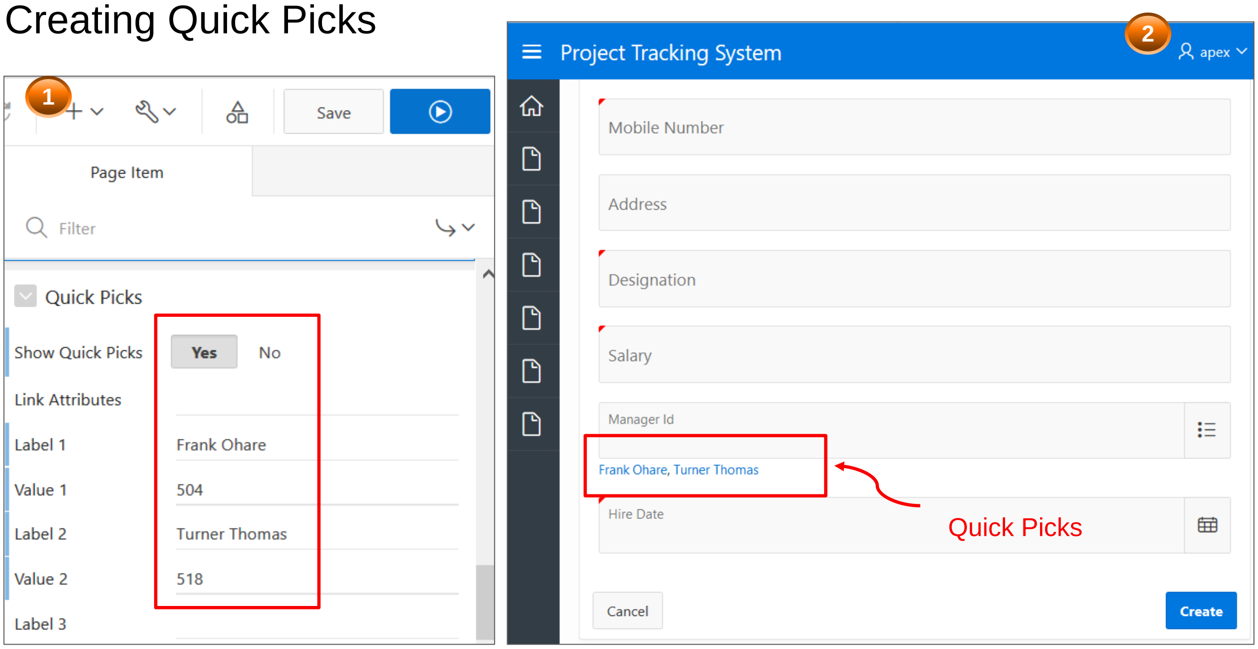
Type Select List

In the previous slide you changed a **Number Field** to a **Select List** item. Now in this slide, you change the **Manager ID** Type from **Select List** to **Popup LOV**.

1. Under Rendering, click **P11_MANAGER_ID** under Page Items.
2. Under **Identification** in Properties Pane on the right side, change **Type** from **Select List** to **Popup LOV**.
3. Click **Save and Run** icon. On running the report, you will find the icon for pop-up LOV beside the Manager ID field. Click it to open the pop-up window with all the Manager names to select from.

Once you select a Manager Name, you can easily get a list of employees reporting to him/her.

Creating Quick Picks



Quick Picks

Show Quick Picks	Yes
Link Attributes	
Label 1	Frank Ohare
Value 1	504
Label 2	Turner Thomas
Value 2	518
Label 3	

Mobile Number

Address

Designation

Salary

Manager Id

Frank Ohare, Turner Thomas

Hire Date

This slide shows Steve updating his *Create Employees* Form (in the *PTS* application) with *Managers ID* as *Quick Picks*. He knows that while filling out the new employee details, the manager name can be easily picked up from the quick picks list. In this example, Steve has added the names of Frank Ohare and Turner Thomas as quick picks for the *Manager ID*.

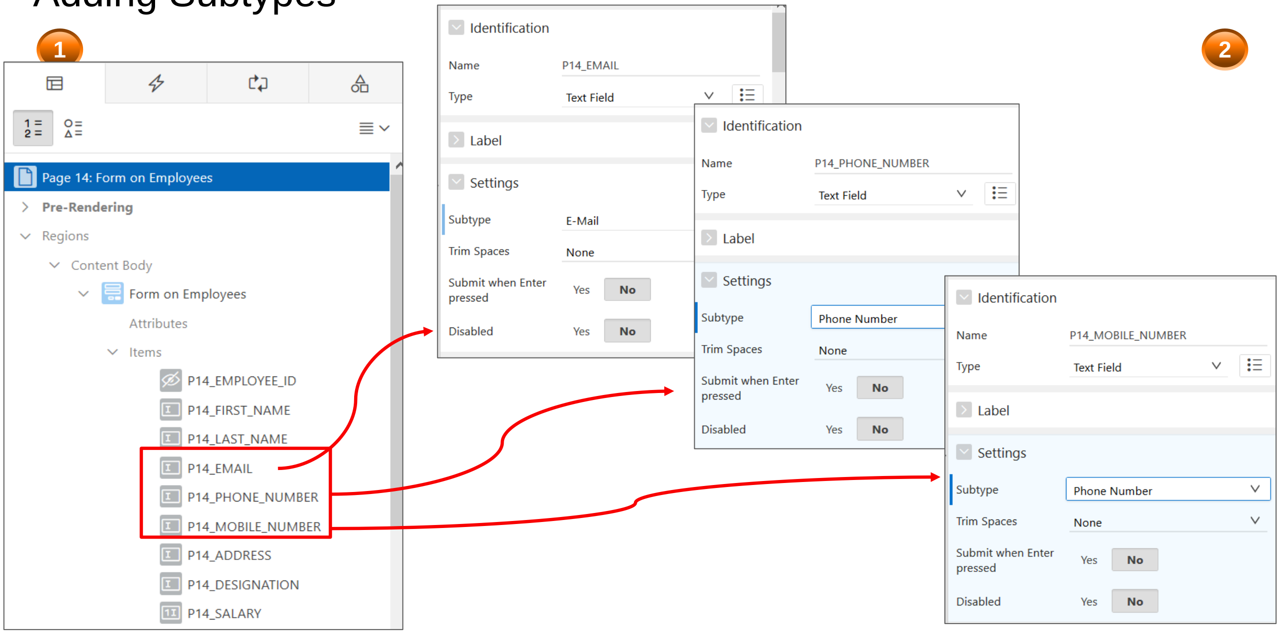
So, what are *Quick Picks*?

Quick picks are links that you display below an item. You can click the quick pick links to enter a value in the item field. You can create up to 10 selections for items that support quick picks, such as text field, number field, select list, and pop-up LOV.

1. To create quick picks, select the item in the Rendering pane of the Page Designer. Here Steve chooses `MANAGER_ID` (in the *Create Employees* form page)
2. In the Property Editor pane, under **Quick Picks**, select **Yes** for **Show Quick Picks**.
3. Enter the label name and value for each quick pick that you want to create (screenshot 1). For example:
   - **Label 1**: `Frank Ohare`
   - **Value 1**: `504`
   - **Label 2**: `Turner Thomas`
   - **Value 2**: `518`
4. Save and run the page to view the created quick picks (screenshot 2).

In the example in the slide, two quick picks are created for the Manager ID item. Click on the Manager Name to see the employees reporting under him/her.

Adding Subtypes



Identification
Name: P14_EMAIL
Type: Text Field
Label
Settings
Subtype: E-Mail
Trim Spaces: None
Submit when Enter pressed: Yes No
Disabled: Yes No

Identification
Name: P14_PHONE_NUMBER
Type: Text Field
Label
Settings
Subtype: Phone Number
Trim Spaces: None
Submit when Enter pressed: Yes No
Disabled: Yes No

Identification
Name: P14_MOBILE_NUMBER
Type: Text Field
Label
Settings
Subtype: Phone Number
Trim Spaces: None
Submit when Enter pressed: Yes No
Disabled: Yes No

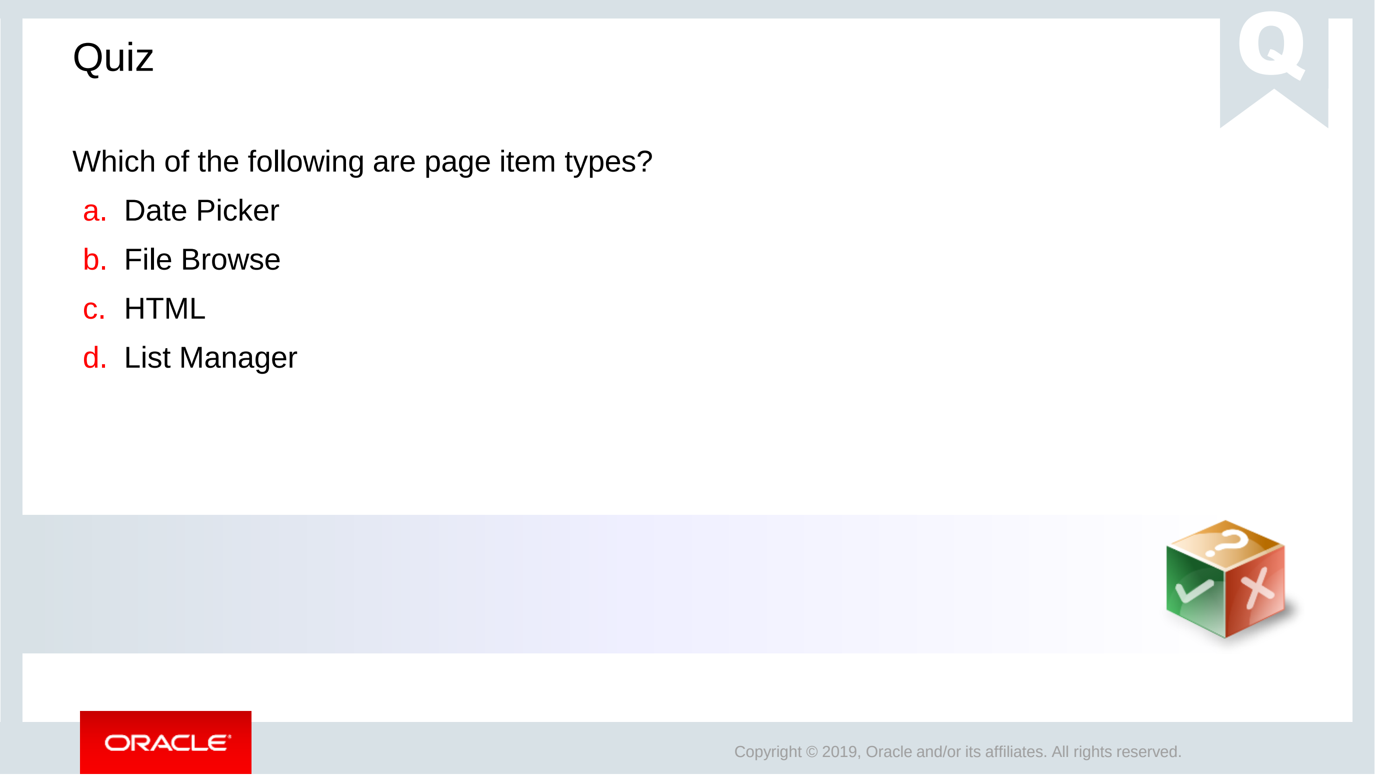
Steve decides to further enhance the user experience. This time he wants to ensure that if his managers are using their mobile devices to work on the forms, the form page must use the appropriate keyboard for email and phone numbers when the relevant values are entered in the form. You will see how he does it by creating a subtype for Email, Phone Number, and Mobile Number fields.

So, you must be wondering what is a *Subtype*? Subtype specifies what kind of text field the page item is. This allows devices with onscreen keyboards to show an optimized keyboard layout specific to the subtype, for easier data input. This HTML5 feature works in modern browsers only. To add subtypes:

1. In the Rendering pane, select the item for which you wish to set up subtype.
2. In the Property Editor pane, select the appropriate subtype for the item from the drop-down menu.

Let's see how Steve creates the subtype for Email, Phone Number, and Mobile Number on the *Form on Employees* page:

1. Open **Form on Employees** in Page Designer view.
2. In the Page Designer view, create a subtype for Email, Phone Number, and Mobile Number fields:
   - Under Rendering, click P14_EMAIL under **Page Items** to see its Properties Pane on right side of the page. Enter the following value:
     - **Settings**: **Subtype**: `Select` *`E-Mail`*
   - Under Rendering, click P14_PHONE_NUMBER under **Page Items** to see its Properties Pane on right side of the page. Enter the following value:
     - **Settings**: **Subtype**: `Select` *`Phone Number`*
   - Under Rendering, click P14_MOBILE_NUMBER under **Page Items** to see its Properties Pane on right side of the page. Enter the following value:
     - **Settings**: **Subtype**: `Select` *`Phone Number`*
3. Click **Run** icon on the page and the form opens. You can see the changes when you enter appropriate values for Email, Phone Number, and Mobile Number. Any wrong value will be displayed as an error message.



Quiz

Which of the following are page item types?

a. Date Picker
b. File Browse
c. HTML
d. List Manager

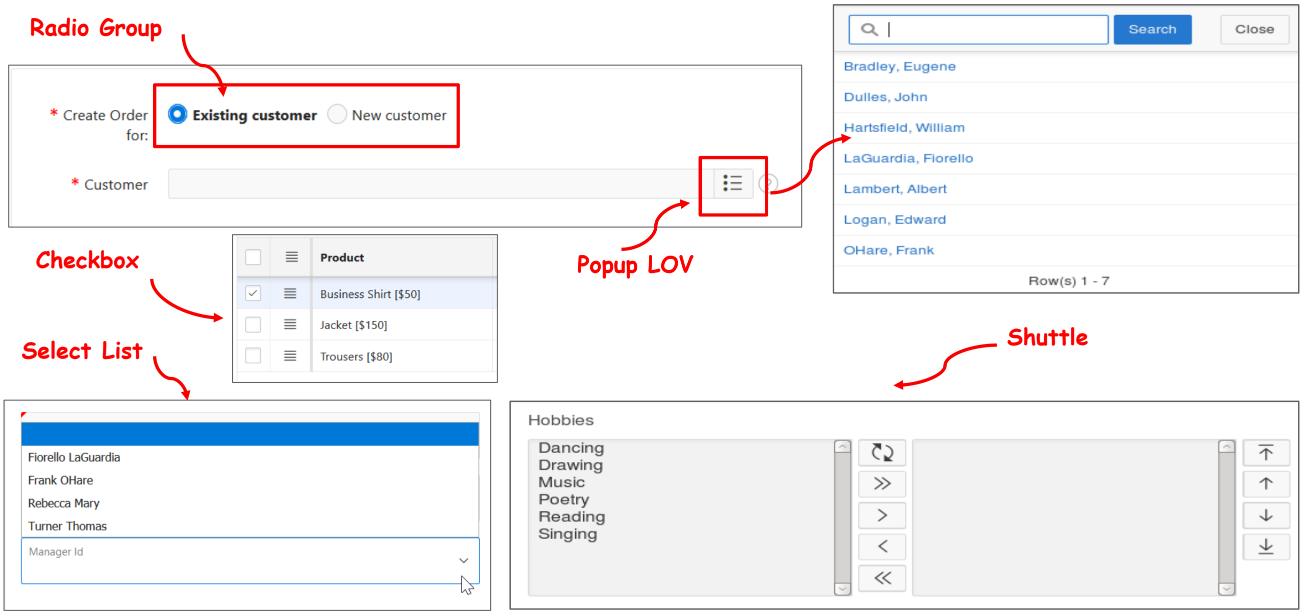
Answer: a, b, d

Lesson Agenda

- Introducing Items
- Using Items
- Creating List of Value (LOV) Type Items
  - What Is an LOV?
  - Accessing the Lists of Values Page
  - Creating a Static LOV
  - Creating a Dynamic LOV
  - Associating an LOV with an Item
  - Creating a Select List Item
  - Creating a Cascading LOV
- Using Buttons

ORACLE

What Is an LOV?



There are different ways you can display a list of items and values—by creating a radio group, or a select list, or a popup LOV, for example. Let’s now go into a little more detail and understand what an LOV is.

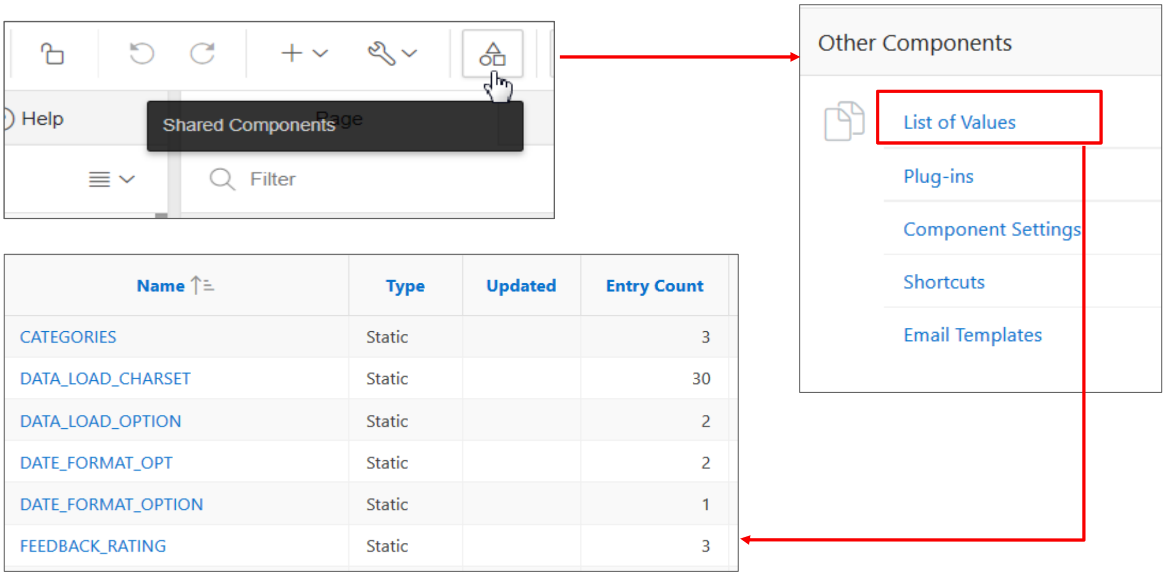
A list of values (LOV) is used to display values for some specific type of page item, such as a radio group, check box, or select list. You can create an LOV while creating the item or create an LOV as a shared component and then reference it in one or more items. An LOV can be either of the following:

- *Static*: Based on a set of predefined display and return values
- *Dynamic*: Based on a SQL query that selects values from tables

There is also a *Cascading* LOV, which is a *Dynamic* LOV, where one item on a page determines the list of values for another item.

You will learn how to create and assign these LOVs in later slides.

Accessing the Lists of Values Page



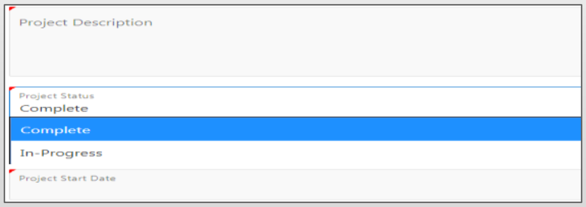
Name	Type	Updated	Entry Count
CATEGORIES	Static		3
DATA_LOAD_CHARSET	Static		30
DATA_LOAD_OPTION	Static		2
DATE_FORMAT_OPT	Static		2
DATE_FORMAT_OPTION	Static		1
FEEDBACK_RATING	Static		3

The LOVs that are created as shared components are listed on the *List of Values* page. To access the *List of Values* page, navigate to the **Shared Components** page for the application. Under Other Components, click **List of Values**. The LOVs that are created for the application are displayed. You can create new LOVs or create a copy of an existing LOV.

Note: Shared component LOVs are also called *named* LOVs.

Creating a Static LOV

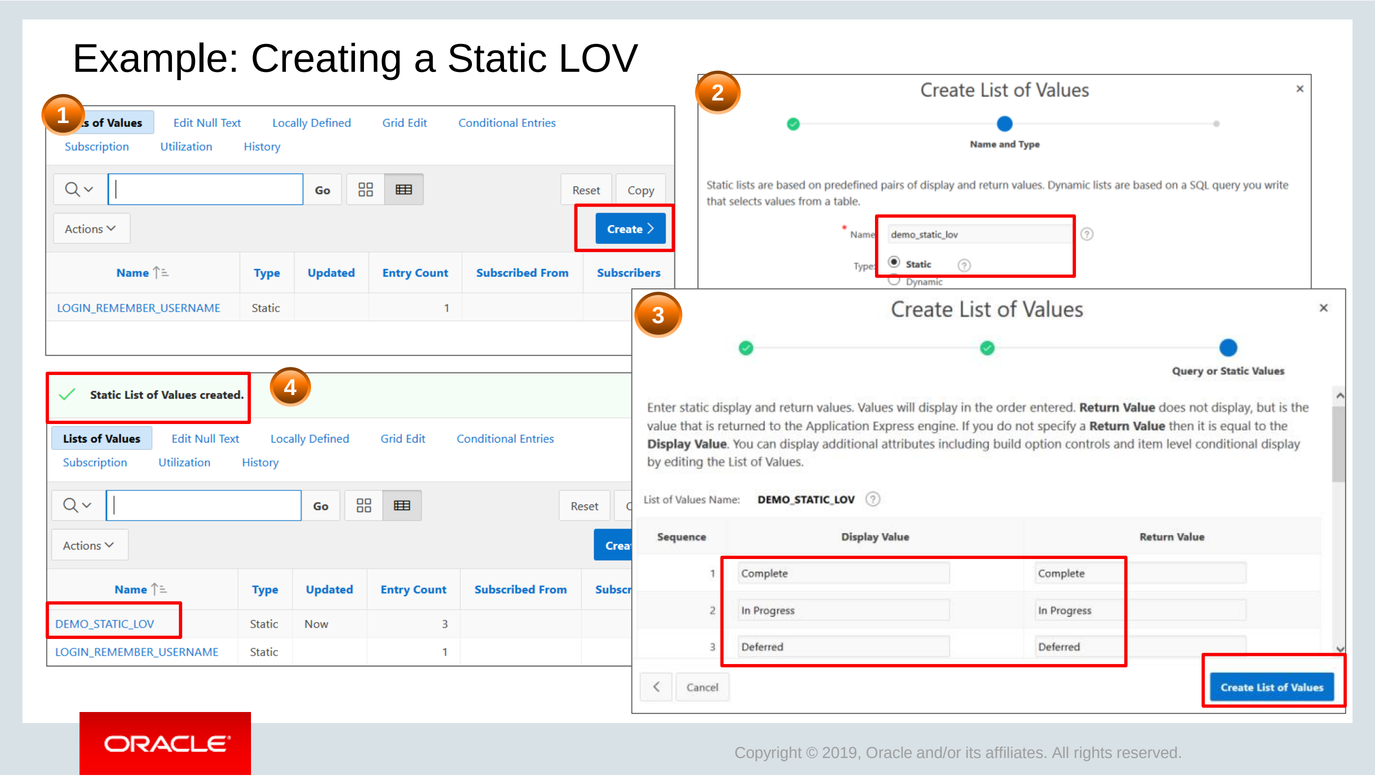
1. In the List of Values page, click `Create`. The steps to navigate to the List of Values page are explained in the previous slide.
2. Select From Scratch and click `Next >`. Note that you can also select the second option to create a copy of an LOV from another application in the same workspace.
3. Enter a name for the LOV. Select `Static` and click `Next >`.
4. Enter the static display and return values. Values are displayed in the order in which they are entered here. The return value is not displayed and is the value returned to the Oracle Application Express engine. In the case where you do not enter a return value, the display value is also the return value.
5. Click `Create List of Values`.



You learned about *Static* LOVs a couple of slides back. Now Steve wants the *Project Status* item in the *Manage Projects* form to have status as either *Complete* or *In-Progress*. Let's see how he does it by creating and assigning a static LOV in the next few slides.

This slide provides an overview of the steps to create a static LOV based on a predefined list of display and return values. New named LOVs are added to the List of Values repository.

Example: Creating a Static LOV



Name	Type	Updated	Entry Count	Subscribed From	Subscribers
LOGIN_REMEMBER_USERNAME	Static		1		

Static lists are based on predefined pairs of display and return values. Dynamic lists are based on a SQL query you write that selects values from a table.

Name: demo_static_lov

Type: ● Static ○ Dynamic

Enter static display and return values. Values will display in the order entered. **Return Value** does not display, but is the value that is returned to the Application Express engine. If you do not specify a **Return Value** then it is equal to the **Display Value**. You can display additional attributes including build option controls and item level conditional display by editing the List of Values.

List of Values Name: **DEMO_STATIC_LOV**

Sequence	Display Value	Return Value
1	Complete	Complete
2	In Progress	In Progress
3	Deferred	Deferred

Static List of Values created.

Name	Type	Updated	Entry Count	Subscribed From	Subscr
DEMO_STATIC_LOV	Static	Now	3		
LOGIN_REMEMBER_USERNAME	Static		1		

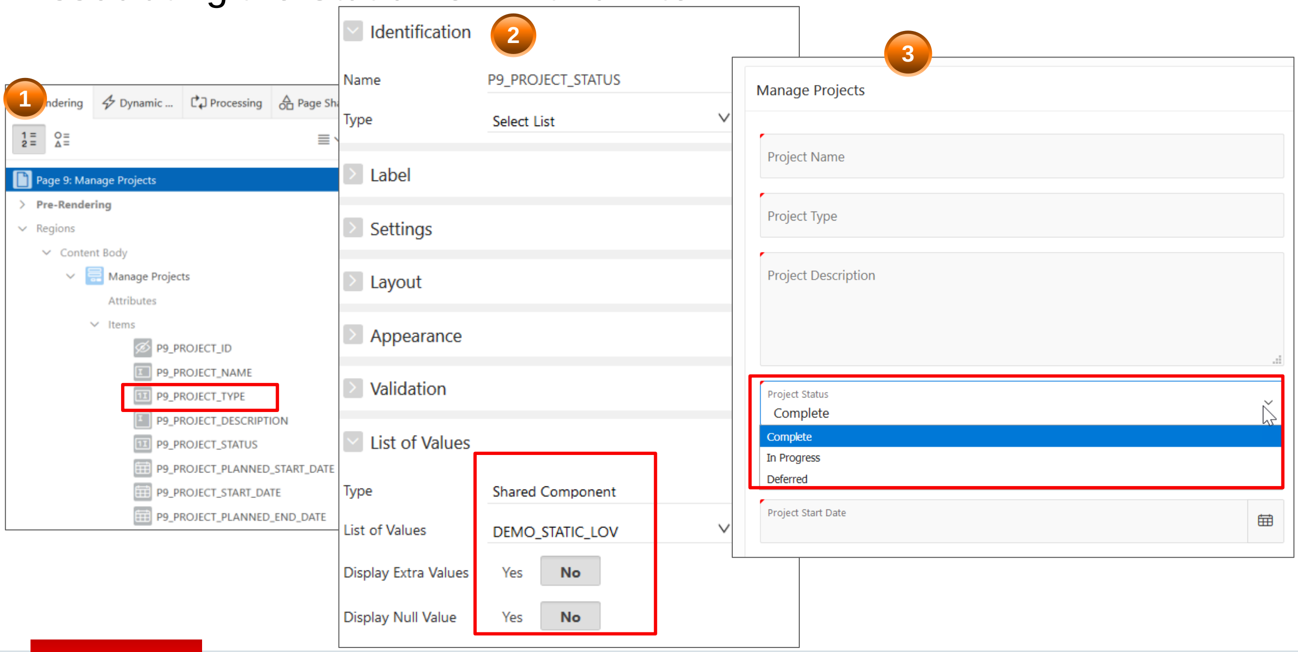
Since Steve wants to create the Project Status as *In-Progress* and *Complete*, let's see how he creates a *Static LOV* to do that.

1. To access the **Lists of Values** page, navigate to the **Shared Components** page for the application. Under **Other Components**, click **Lists of Values**.
2. In the **List of Values** page, click **Create** (screenshot 1).
3. Select **From Scratch** and click **Next** >.
4. Enter a name `DEMO_STATIC_LOV` for the **LOV**. Select **Static** and click **Next** > (screenshot 2).
5. Enter the static display and return values (screenshot 3).
   - **Display:** *Complete*, **Return:** *Complete*
   - **Display:** *In-Progress*, **Return:** *In-Progress*
   - **Display:** *Deferred*, **Return:** *Deferred*
6. Click Create **List of Values** (screenshot 4).

After you add a *Static* LOV to the repository, you can create a check box, radio group, select list, or pop-up list item and reference the LOV to that item.

Steve later associates this *Static* LOV in his *Project Status* item. You will know how to do this in the next slide.

Associating the Static LOV with an Item



Name: P9_PROJECT_STATUS

Type: Select List

List of Values

Type: Shared Component

List of Values: DEMO_STATIC_LOV

Display Extra Values: Yes [✓ No]

Display Null Value: Yes [✓ No]

Project Status: Complete

- Complete
- In Progress
- Deferred

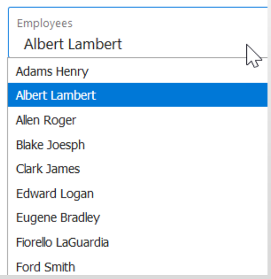
In this slide, Steve associates the *Static* LOV that he just created to the item `PROJECT_STATUS` in the *Manage Projects* form. This way, when he goes to the *Manage Projects* form, he can choose the status as *Complete, Deferred,* or *In-Progress*.

You can associate a named LOV with an item that can accept a list of values. To associate an LOV to an item, perform the following steps:

1. In the Rendering tab of the Page Designer, select the item. Here Steve selects *Project Status* item.
2. In the Property Editor pane, ensure that the display type is an LOV type item by checking the Type field. You can change the display type, if required. In this example, the *Product Status* field is changed to **Select List**.
3. In the Property Editor pane, scroll down to List of Values. Select **Shared Component** for Type and your LOV list for List of Values. In this example, Steve selects `DEMO_STATIC_LOV`.
4. Select the following values:
   - **Display Extra Values**: Select **No**
   - **Display Null Value**: Select **No**
5. Click **Save**. Run the page to check whether the item displays the list of values in *Manage Projects* form. You see that a select list having only *Complete* or *In-progress* values is provided for **Project Status** field on the form.

Creating a Dynamic LOV

1. In the List of Values page, click `Create`.
2. Select `From Scratch` and click `Next >`.
3. Enter a name for the LOV. Select `Dynamic` and click `Next >`.
4. The first column returns the values to be displayed in the items list.
5. The second column gives the value that is returned to the Oracle Application Express engine when the display value is selected. Enter a SQL query that returns two columns.
6. If the display and return columns are the same, or if a column includes a function or operator, you must use column aliases in the query.
7. Click `Create List of Values`.



Employees

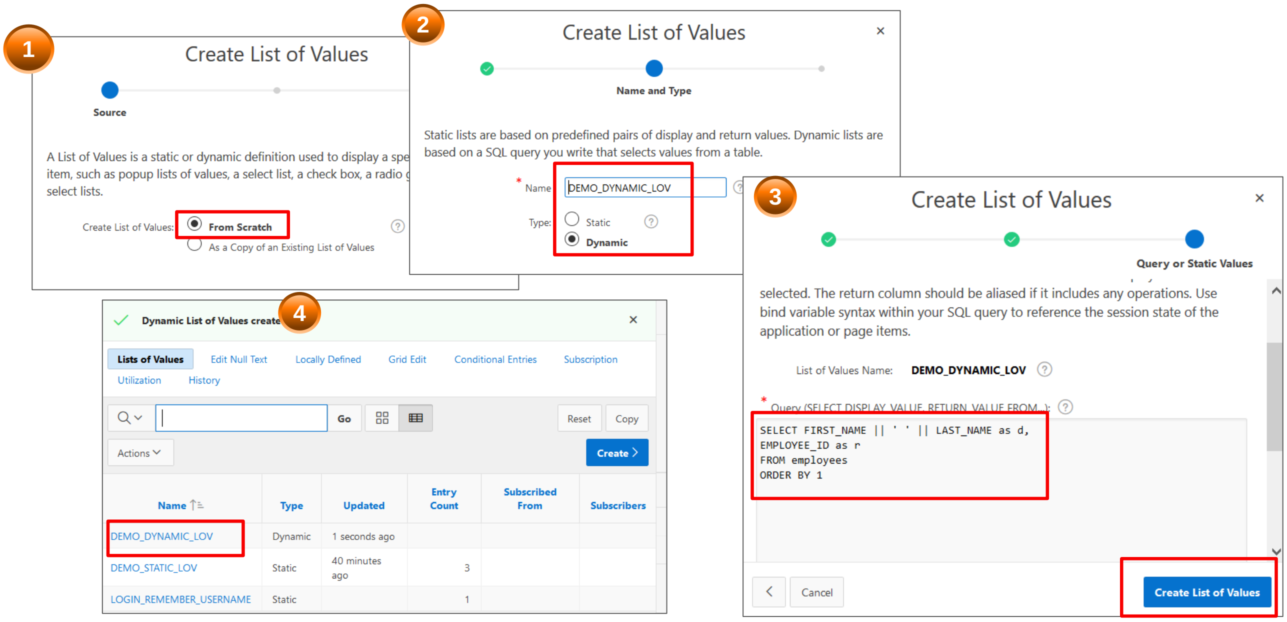
Albert Lambert

Adams Henry
Albert Lambert
Allen Roger
Blake Joesph
Clark James
Edward Logan
Eugene Bradley
Fiorello LaGuardia
Ford Smith

Now that you have created a *Static* LOV and assigned it to an item, let's try creating a *Dynamic* LOV and assigning it too. Steve wants to display the list of all the employees belonging to his `EMPLOYEES` database. This will be helpful for Stella and other managers as they would get to see a list of all the employees in the *Employees* tab and can easily distribute the projects. Let's see how Steve does it by creating a *Dynamic* LOV.

This slide provides an overview of the steps to create a *Dynamic* LOV. Dynamic LOVs are based on SQL queries that are executed at run time and select values from tables or views.

Example: Creating a Dynamic LOV



Create List of Values

Source

A List of Values is a static or dynamic definition used to display a spe item, such as popup lists of values, a select list, a check box, a radio g select lists.

Create List of Values: ● From Scratch ○ As a Copy of an Existing List of Values

Create List of Values

Name and Type

Static lists are based on predefined pairs of display and return values. Dynamic lists are based on a SQL query you write that selects values from a table.

* Name: DEMO_DYNAMIC_LOV

Type: ○ Static ● Dynamic

Create List of Values

Query or Static Values

selected. The return column should be aliased if it includes any operations. Use bind variable syntax within your SQL query to reference the session state of the application or page items.

List of Values Name: DEMO_DYNAMIC_LOV

* Query (SELECT DISPLAY_VALUE, RETURN_VALUE FROM...)

```
SELECT FIRST_NAME || ' ' || LAST_NAME as d,
EMPLOYEE_ID as r
FROM employees
ORDER BY 1
```

Dynamic List of Values create

Lists of Values | Edit Null Text | Locally Defined | Grid Edit | Conditional Entries | Subscription | Utilization | History

Name	Type	Updated	Entry Count	Subscribed From	Subscribers
DEMO_DYNAMIC_LOV	Dynamic	1 seconds ago			
DEMO_STATIC_LOV	Static	40 minutes ago	3		
LOGIN_REMEMBER_USERNAME	Static		1		

ORACLE

This slide provides an example of the steps to create a *Dynamic* LOV.

1. Navigate to the **Shared Components** page for the application. Under **Other Components**, click **Lists of Values**.
2. In the **List of Values** page, click **Create**.
3. Select **From Scratch** and click **Next >**.
4. Enter a name `DEMO_DYNAMIC_LOV` for the **LOV**. Select **Dynamic** and click **Next >**.
5. Enter the query (since Steve wants to display the list of all the employees belonging to his `EMPLOYEES` database) :

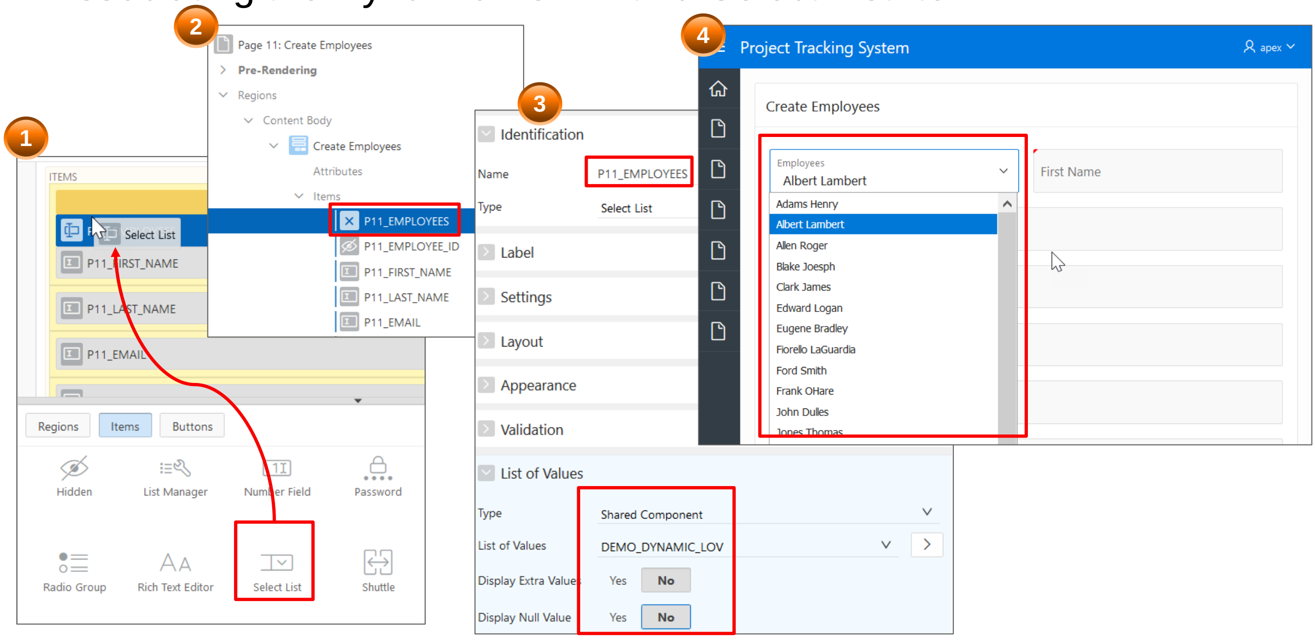
```
SELECT FIRST_NAME || ' ' || LAST_NAME as d,
EMPLOYEE_ID as r
FROM employees
ORDER BY 1
```

6. Click **Create List of Values**.

After you add a *Dynamic* LOV to the repository, you can create a check box, radio group, select list, or pop-up list item and reference the LOV to that item.

Steve later associates this *Dynamic* LOV in his `EMPLOYEES` item. You will know how to do this in the next slide.

Associating the Dynamic LOV with a Select List Item



1

ITEMS

Select List

P11_FIRST_NAME

P11_LAST_NAME

P11_EMAIL

Regions | Items | Buttons

Hidden, List Manager, Number Field, Password, Radio Group, Rich Text Editor, Select List, Shuttle

2

Page 11: Create Employees
- Pre-Rendering
- Regions
  - Content Body
    - Create Employees
      - Attributes
      - Items
        - P11_EMPLOYEES
        - P11_EMPLOYEE_ID
        - P11_FIRST_NAME
        - P11_LAST_NAME
        - P11_EMAIL

3

Identification

Name: P11_EMPLOYEES

Type: Select List

Label

Settings

Layout

Appearance

Validation

List of Values

Type: Shared Component

List of Values: DEMO_DYNAMIC_LOV

Display Extra Values: Yes [✓ No]

Display Null Value: Yes [✓ No]

4

Project Tracking System

Create Employees

Employees: Albert Lambert

First Name

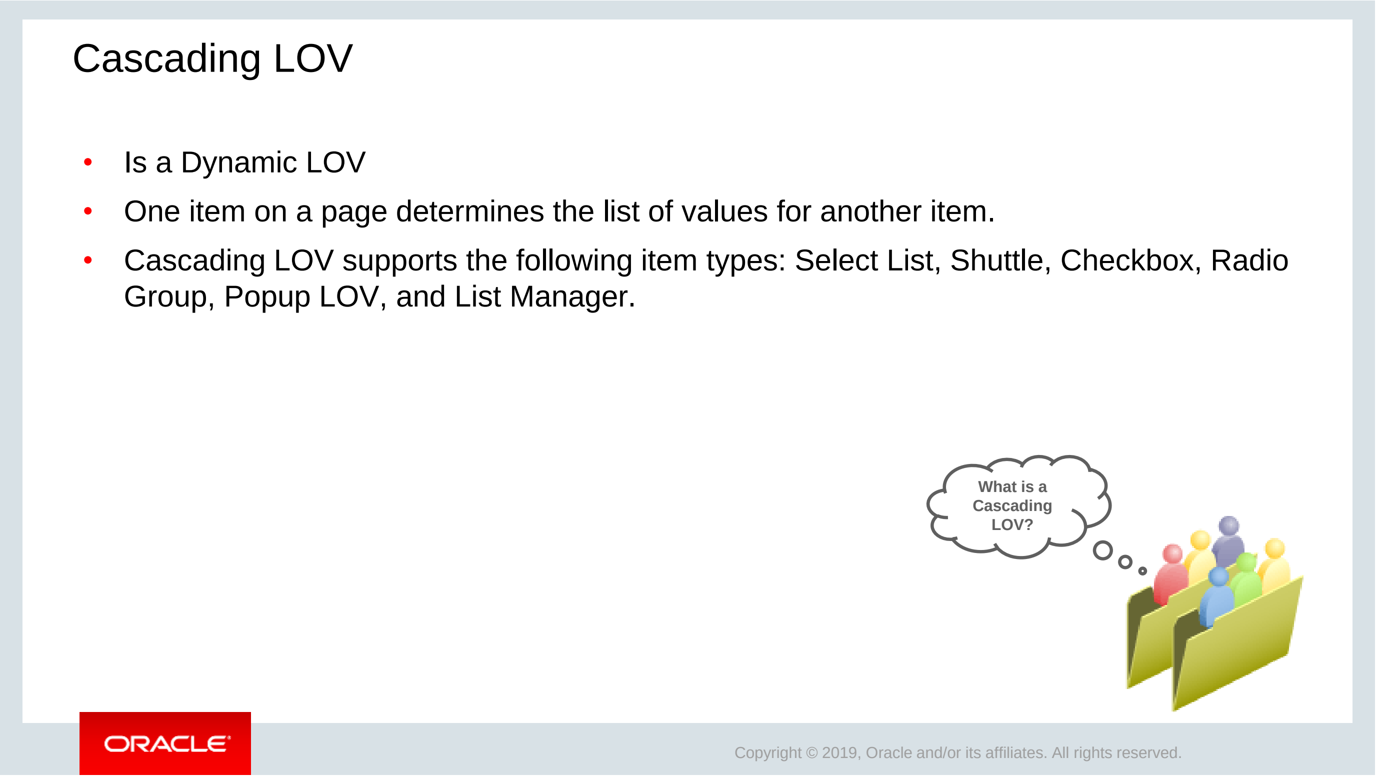
Adams Henry
Albert Lambert
Allen Roger
Blake Joesph
Clark James
Edward Logan
Eugene Bradley
Fiorello LaGuardia
Ford Smith
Frank OHare
John Dulles
Jones Thomas

Now that you know how to create a *Dynamic* LOV, let’s see how you can assign it to either a new item or to an existing item.

In this slide, Steve creates a **Select List** item called `EMPLOYEES` in his *Create Employees* form and later assigns the Dynamic LOV that he created in the slide before. The main purpose is to display the list of employees in the *Employees* tab.

The steps to create a **Select List** item is very similar to creating a **Date Picker** item that was discussed earlier in this lesson. Just like the date picker, you can create a select list item in two ways: from the Rendering Pane or from the Items gallery. In this example, the latter is used.

1. Choose the **Select List** item from the Items gallery. Drag it from the gallery to the Grid Layout of the page under the region where you want to create the item (screenshot 1).
2. In the Identification section of the Property Editor pane, enter a name for the item. As a best practice, use the format `P<`*n*`>_<ITEM_NAME>` to name the item. In this example, the item is named `P<`*n*`>_EMPLOYEES`. Click the Label field. Notice that the value of the Label field is automatically populated as soon as you click it. Scroll down (screenshot 3).
3. Under List of Values, select **Type** as *Shared Components***.** In the **List of Values** field, select `DEMO_DYNAMIC` (screenshot 2).
4. Select the following (screenshot 3):
   - **Display Extra Values**: Select **No.**
   - **Display Null Value**: Select **No.**
5. Run the page to test if the select list is working (screenshot 4).



Cascading LOV

- Is a Dynamic LOV
- One item on a page determines the list of values for another item.
- Cascading LOV supports the following item types: Select List, Shuttle, Checkbox, Radio Group, Popup LOV, and List Manager.

With many new projects coming in, it is difficult for the Project Mangers to properly manage multiple types of projects. Therefore, Steve wants to use the *Cascading* LOV feature of Oracle Application Express to display the projects belonging to a particular project type.

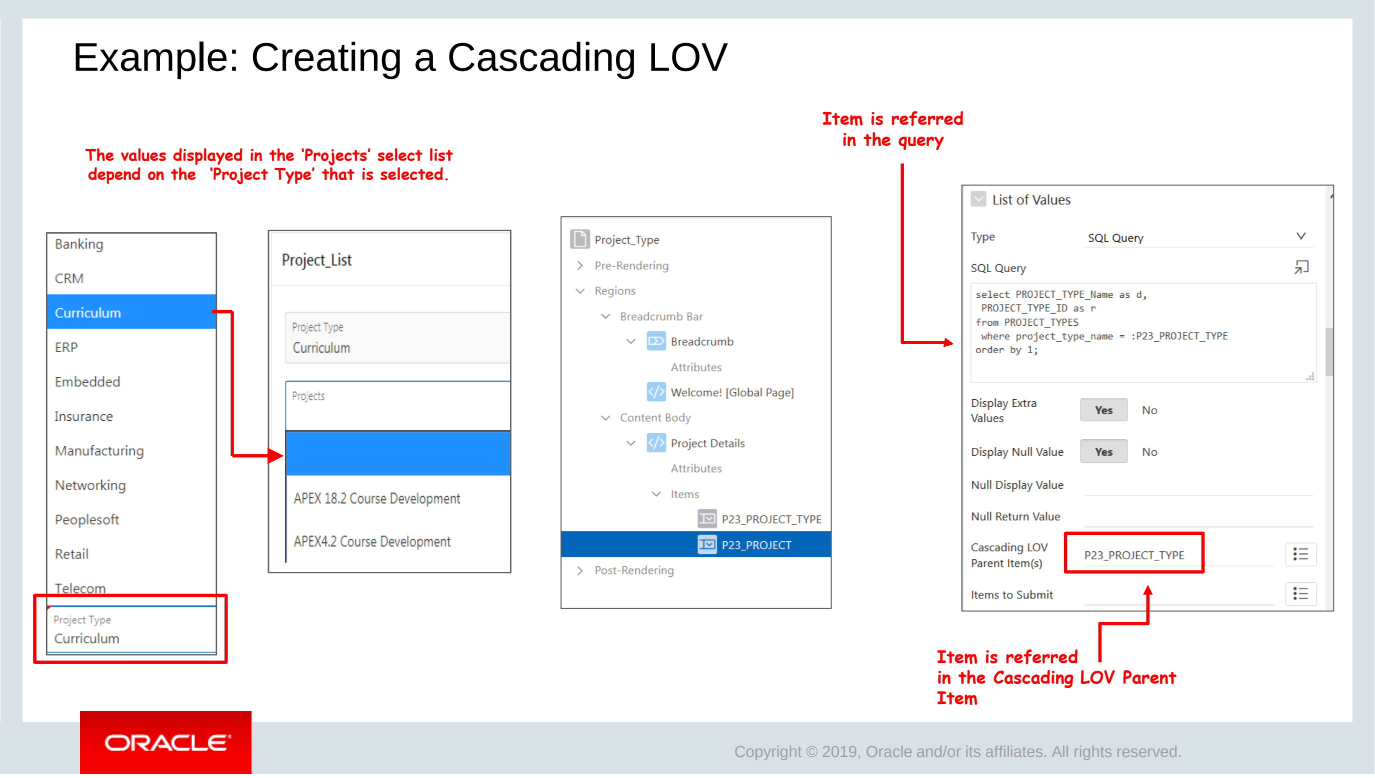
In the next few slides, let's see how Steve uses *Cascading* LOV to help Stella and the other managers handle project assignments efficiently.

So, what is a *Cascading* LOV?

A *Cascading* LOV is a *dynamic* LOV that references another page item for its list of values. You create a *cascading* LOV by creating two items. To populate the second item, you specify a Cascading LOV Parent Item when running the *Create Item Wizard* or by editing the item's attributes.

You can define a *cascading* LOV while creating the item. Before creating a cascading LOV, you must first create the item that is referred to.

In the next slide, you will be creating a *cascading* LOV to display the projects belonging to a particular project type.



Example: Creating a Cascading LOV

The values displayed in the 'Projects' select list depend on the 'Project Type' that is selected.

Item is referred in the query

Item is referred in the Cascading LOV Parent Item

Type: SQL Query

SQL Query:

```
select PROJECT_TYPE_Name as d,
 PROJECT_TYPE_ID as r
from PROJECT_TYPES
 where project_type_name = :P23_PROJECT_TYPE
order by 1;
```

Display Extra Values: Yes No

Display Null Value: Yes No

Null Display Value

Null Return Value

Cascading LOV Parent Item(s): P23_PROJECT_TYPE

Items to Submit

In this slide, a *Cascading* LOV displays the projects belonging to a particular project type (for example, *Curriculum*). Before creating a cascading LOV, you must first create the item that is referred to.

In this example, Steve has first created a Select List item called **PROJECT_TYPE** with the following SQL Query:

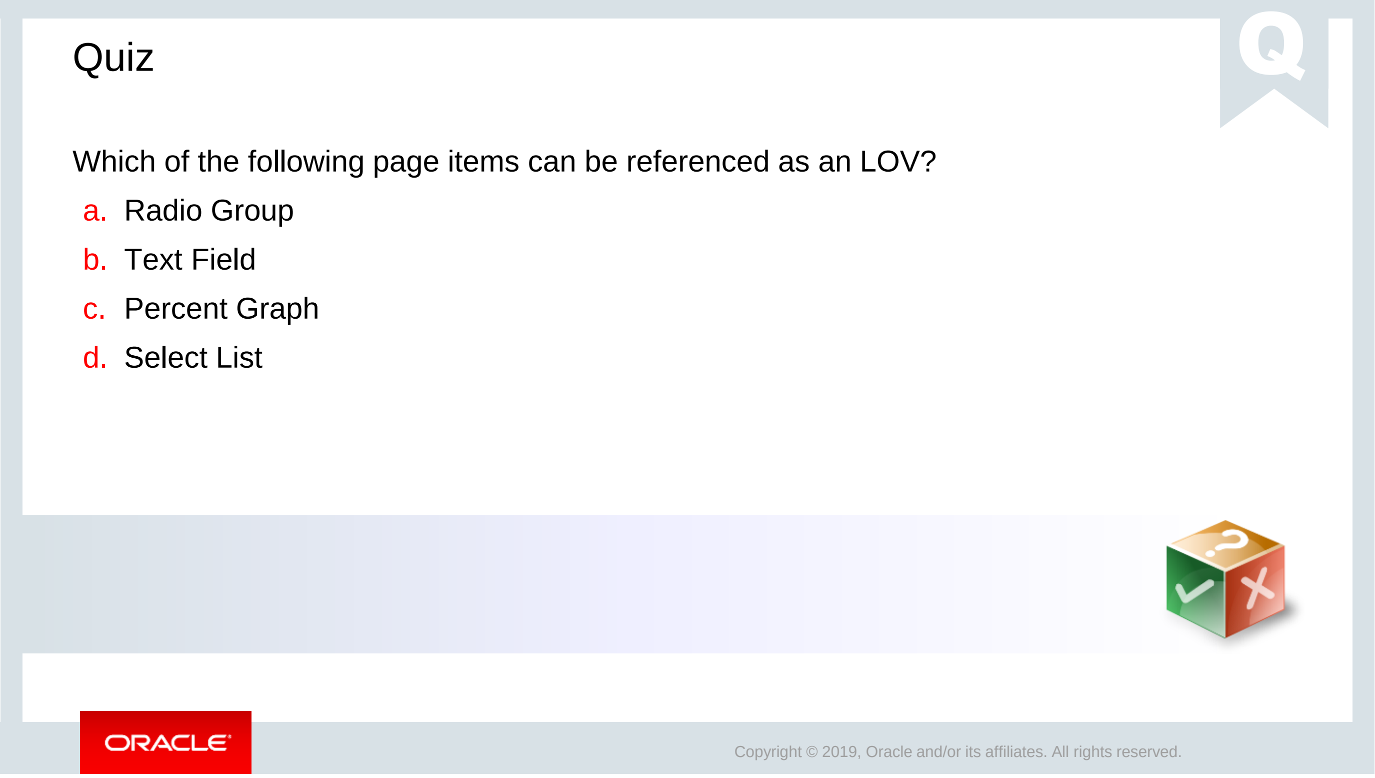
```
select PROJECT_TYPE_Name as d,
 PROJECT_TYPE_ID as r
from PROJECT_TYPES
order by 1;
```

Now, let's see how he defines a cascading LOV item:

1. Navigate to the appropriate page definition. Create the item or select the existing item in the Rendering tab. Here Steve creates another Select List item called **PROJECTS**.
2. In the Property Editor, scroll down to the **List of Values** tab.
3. Enter the following:
   - **List of Values**: **Type**: *SQL Query*
   - Modify the SQL query to include the referred item in the WHERE clause. Since Steve wants to sort the Project Names on the basis of the Project Type, he enters the following SQL Query (in the code text editor):

```
select PROJECT_Name as d,
PROJECT_ID as r
from PROJECTS
where project_type_ID = :P23_PROJECT_TYPE
order by 1;
```

4. Click the *Up* arrow for the Cascading LOV Parent Item(s) field and select the item that you want to refer in the SQL query (in this example: `P23_PROJECT_TYPE`).
5. Save and run the page to confirm that the items are populated as required.

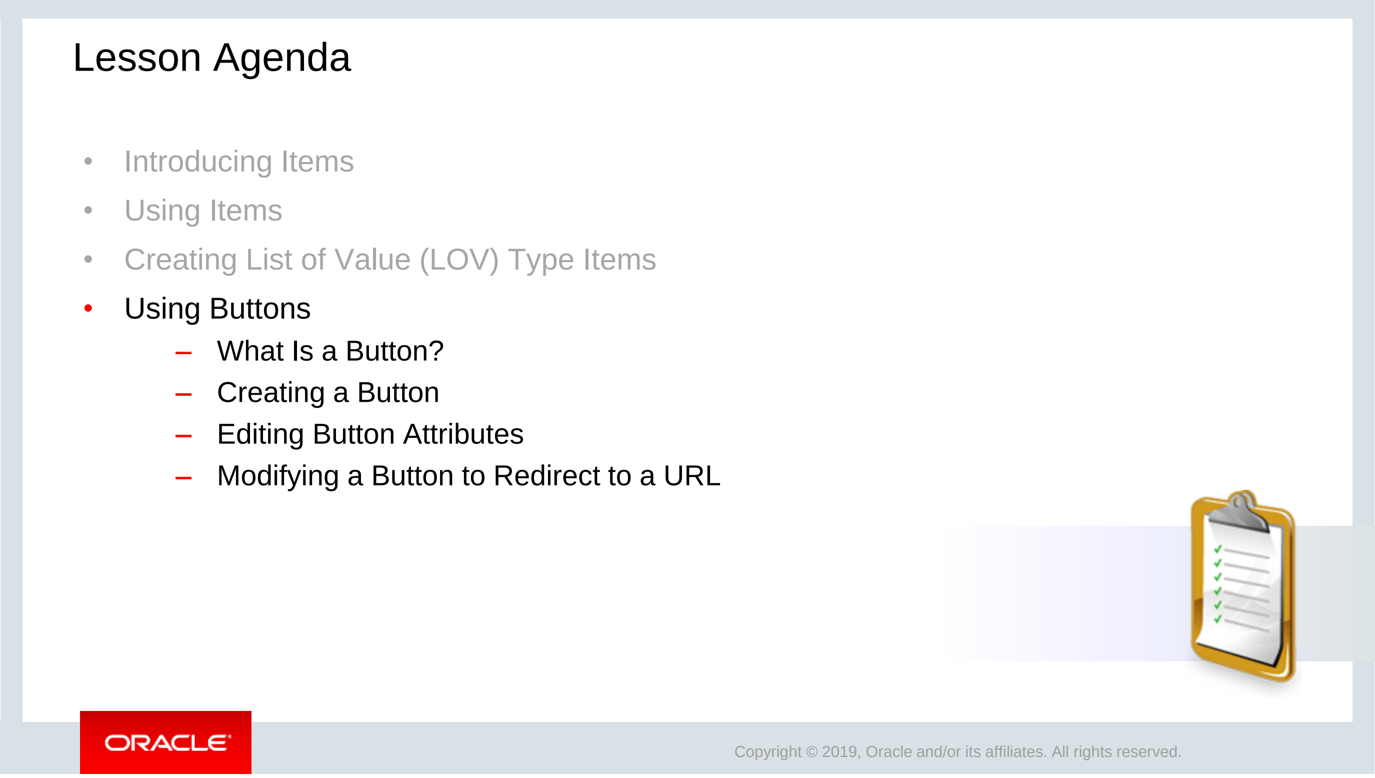


Quiz

Which of the following page items can be referenced as an LOV?

a. Radio Group
b. Text Field
c. Percent Graph
d. Select List

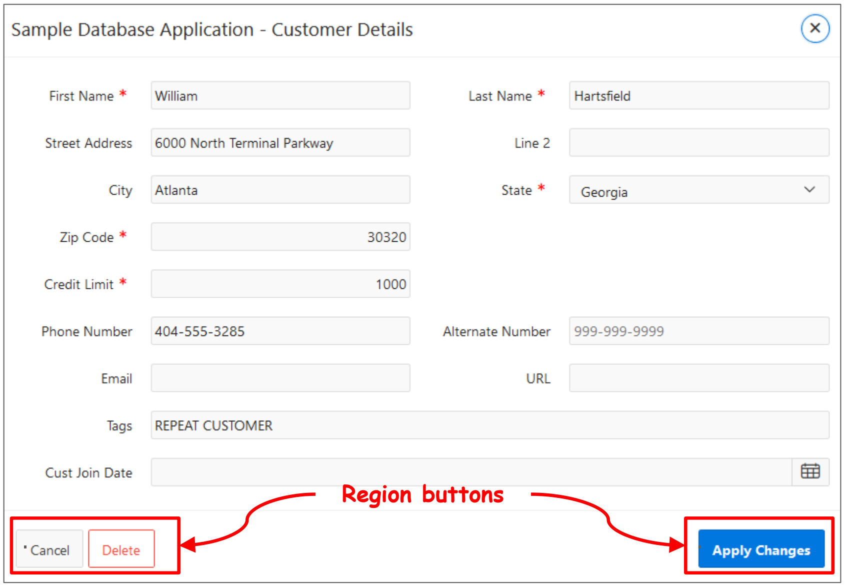
Answer: a and d



Lesson Agenda

- Introducing Items
- Using Items
- Creating List of Value (LOV) Type Items
- Using Buttons
  - What Is a Button?
  - Creating a Button
  - Editing Button Attributes
  - Modifying a Button to Redirect to a URL

What Is a Button?



Sample Database Application - Customer Details

First Name *	William	Last Name *	Hartsfield
Street Address	6000 North Terminal Parkway	Line 2	
City	Atlanta	State *	Georgia
Zip Code *	30320		
Credit Limit *	1000		
Phone Number	404-555-3285	Alternate Number	999-999-9999
Email		URL	
Tags	REPEAT CUSTOMER		
Cust Join Date			

Region buttons

Do you see the *Cancel*, *Delete*, and *Apply Changes* buttons on the *Customer Details* form above?

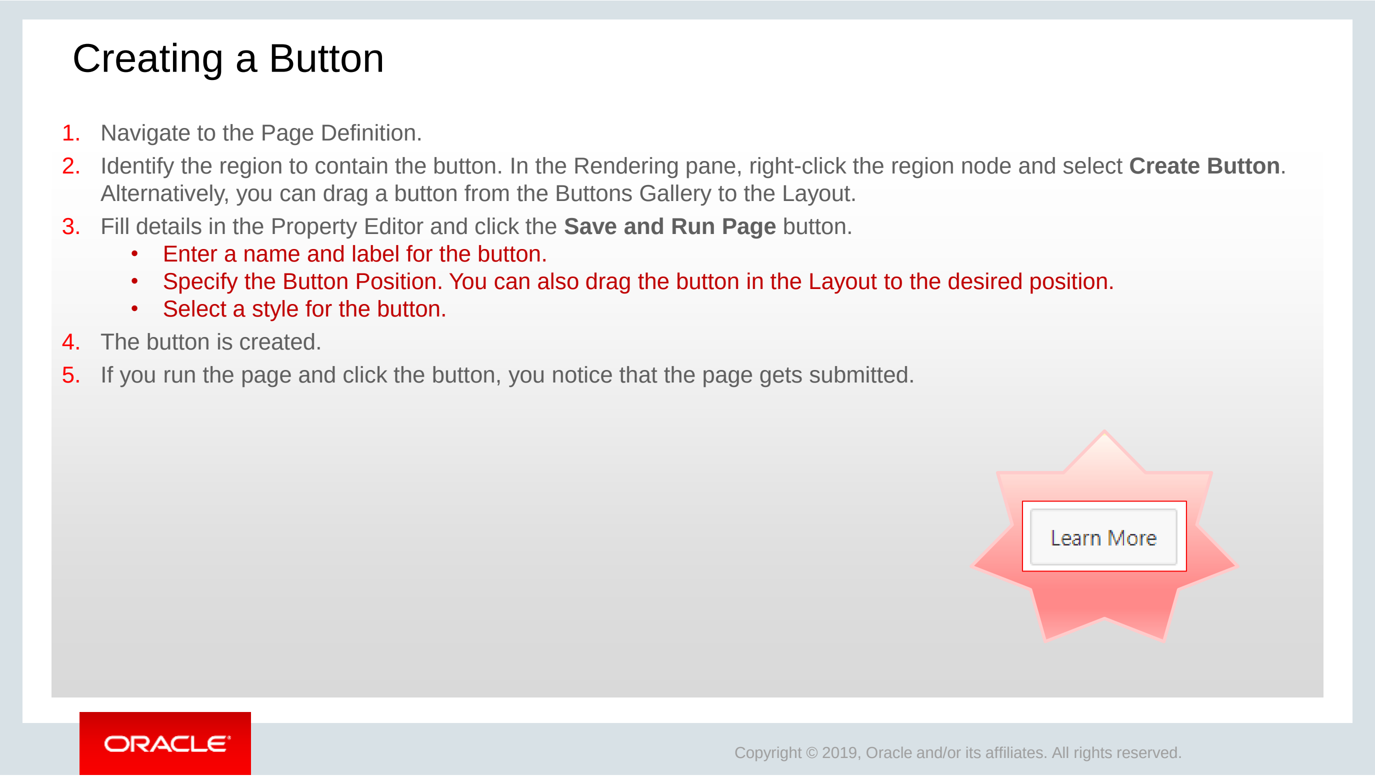
You learned about *Buttons* in the lesson on *Working with Pages and Regions*. We also mentioned this briefly when we were talking about the *Items* tab in one of the initial slides in this lesson.

So, what is a *Button*?

A *Button* is an interface element that is used to either submit a page or navigate to another page or URL. You can create a button that is placed next to other page items. When you use wizards to create page components such as reports and forms, some buttons (such as Cancel, Save, Create, and Delete) are automatically created.

In this lesson, you learn how to create a region button named *Learn More* in the *Form on Employees* which, when clicked, submits the page items and displays a report region. In a later lesson on *Understanding Session State*, you also create a button named *Session Demo* button, which clears the cache for the items on a page and redirects to another page.

You will learn more about how to create buttons in the next few slides.



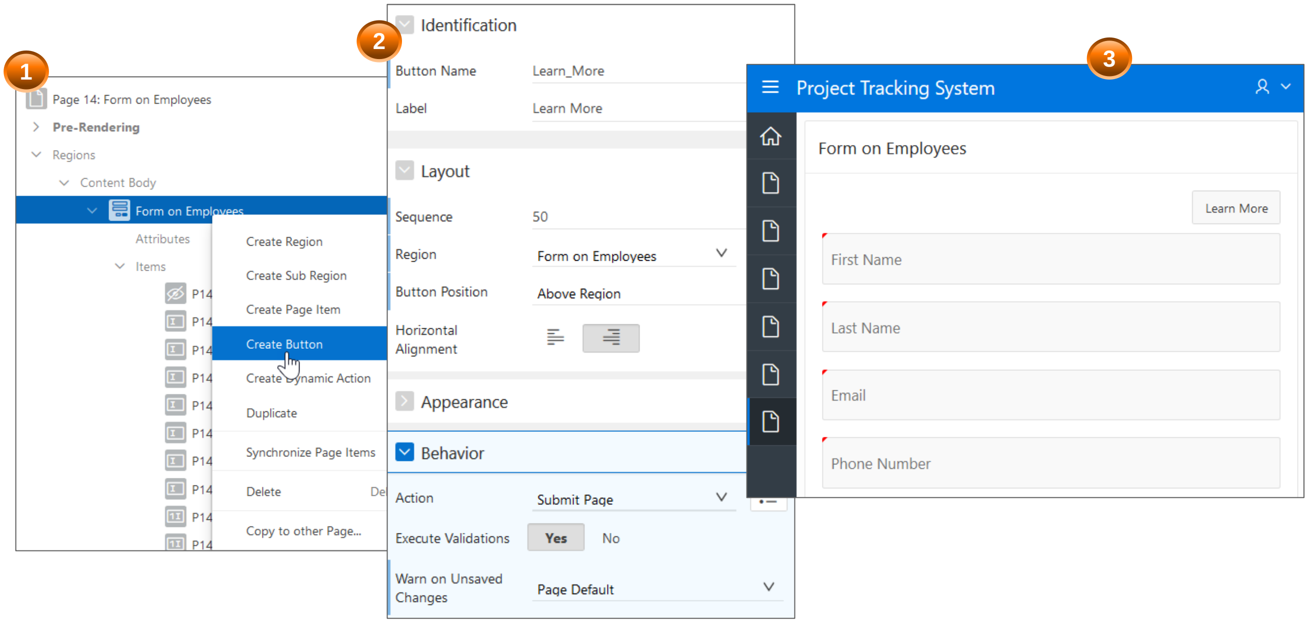
Creating a Button

1. Navigate to the Page Definition.
2. Identify the region to contain the button. In the Rendering pane, right-click the region node and select **Create Button**. Alternatively, you can drag a button from the Buttons Gallery to the Layout.
3. Fill details in the Property Editor and click the **Save and Run Page** button.
   - Enter a name and label for the button.
   - Specify the Button Position. You can also drag the button in the Layout to the desired position.
   - Select a style for the button.
4. The button is created.
5. If you run the page and click the button, you notice that the page gets submitted.

Now, Steve wants to further enhance his user interface, and this time he wants to add buttons to his *Form on Employees*. You will see how he creates a button on this form in the next slide.

This slide provides an overview of the steps to create a new button in your application page.

Example: Creating a Button



Button Name: Learn_More
Label: Learn More
Sequence: 50
Region: Form on Employees
Button Position: Above Region
Action: Submit Page
Execute Validations: Yes No
Warn on Unsaved Changes: Page Default

In an earlier lesson on *Managing Forms*, Steve had created this *Form on Employees* along with the *Employees List View* report while developing a form for Stella to use from her mobile interface.

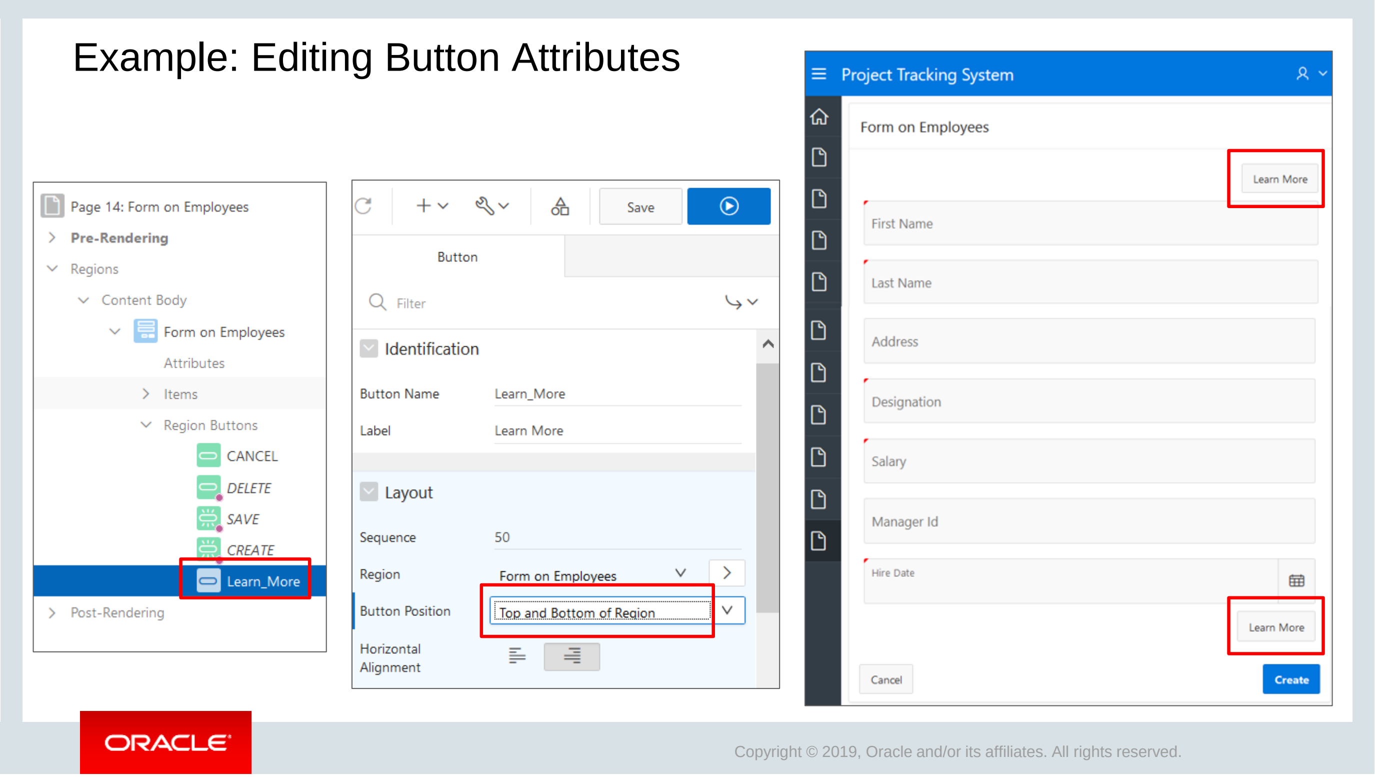
In this slide, Steve creates a *Learn More* button on the *Form on Employees*. Let's see how.

1. Open **Form on Employees** in Page Designer.
2. In the Rendering pane, select *FORM on Employees* under **Regions > Content Body**; right-click and select **Create Button**. Alternatively, you can drag a button from the Buttons Gallery to the Layout.
3. Enter the following:
   - **Identification: Name**: Enter `Learn_More`
   - **Identification**: **Label**: By default `Learn More`
   - **Layout: Button Position**: Select `Above Region`
   - **Action**: By default it is **Submit.** Retain it as it is. We would later modify this.
4. Click Save. The button is created

After the button is created, and Steve runs the *Form on Employees* page and clicks the button, the page gets re-directed to the *Employees List View* report.

In the next few slides, you will be editing/modifying the button attributes.

Example: Editing Button Attributes



Button Name: Learn_More

Label: Learn More

Sequence: 50

Region: Form on Employees

Button Position: Top and Bottom of Region

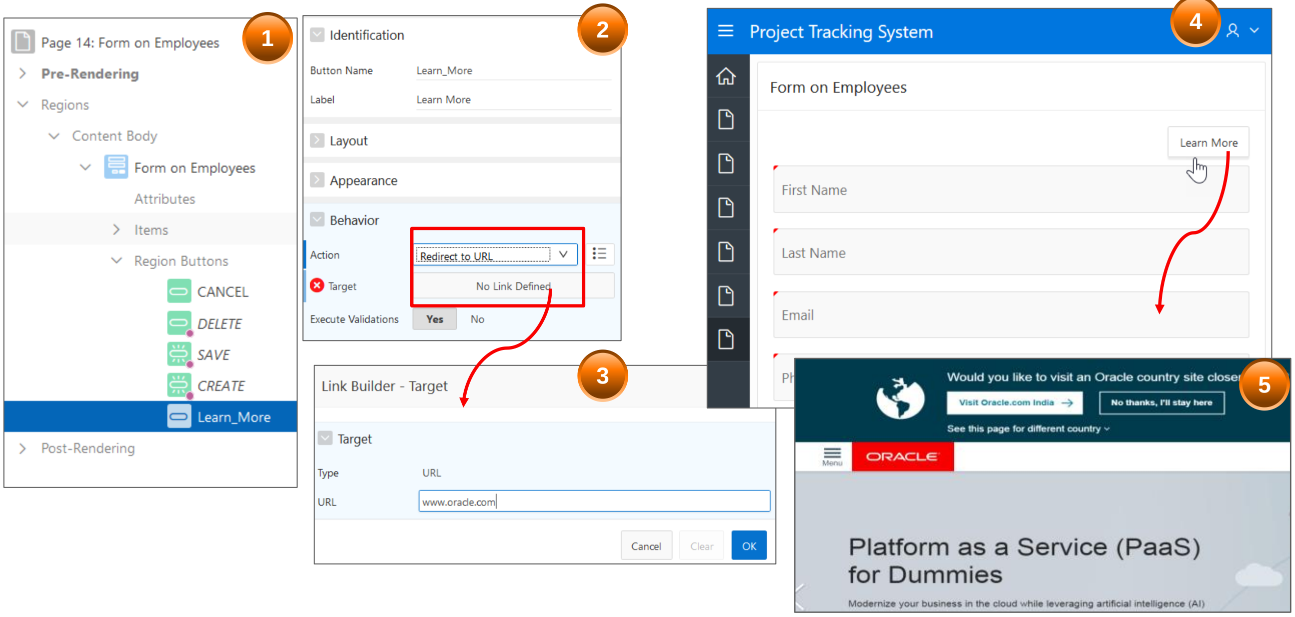
ORACLE

Now that you have created a button, let’s try editing its attributes.

To edit the attributes of a button, select the button in the Rendering pane of the Page Designer. As soon as you select the button, the Property Editor (the right pane) loads the attributes of the button. You can then edit the attributes from this pane.

In this slide, you change the Button Position to **Top and Bottom of Region** from **Below Region**. Save and run the page to verify that the changes to the attributes have been committed.

Modifying a Button to Redirect to a URL



Page 14: Form on Employees
- Pre-Rendering
- Regions
  - Content Body
    - Form on Employees
      - Attributes
      - Items
      - Region Buttons
        - CANCEL
        - DELETE
        - SAVE
        - CREATE
        - Learn_More
- Post-Rendering

Identification

Button Name: Learn_More

Label: Learn More

Behavior

Action: Redirect to URL

Target: No Link Defined

Execute Validations: Yes No

Link Builder - Target

Target

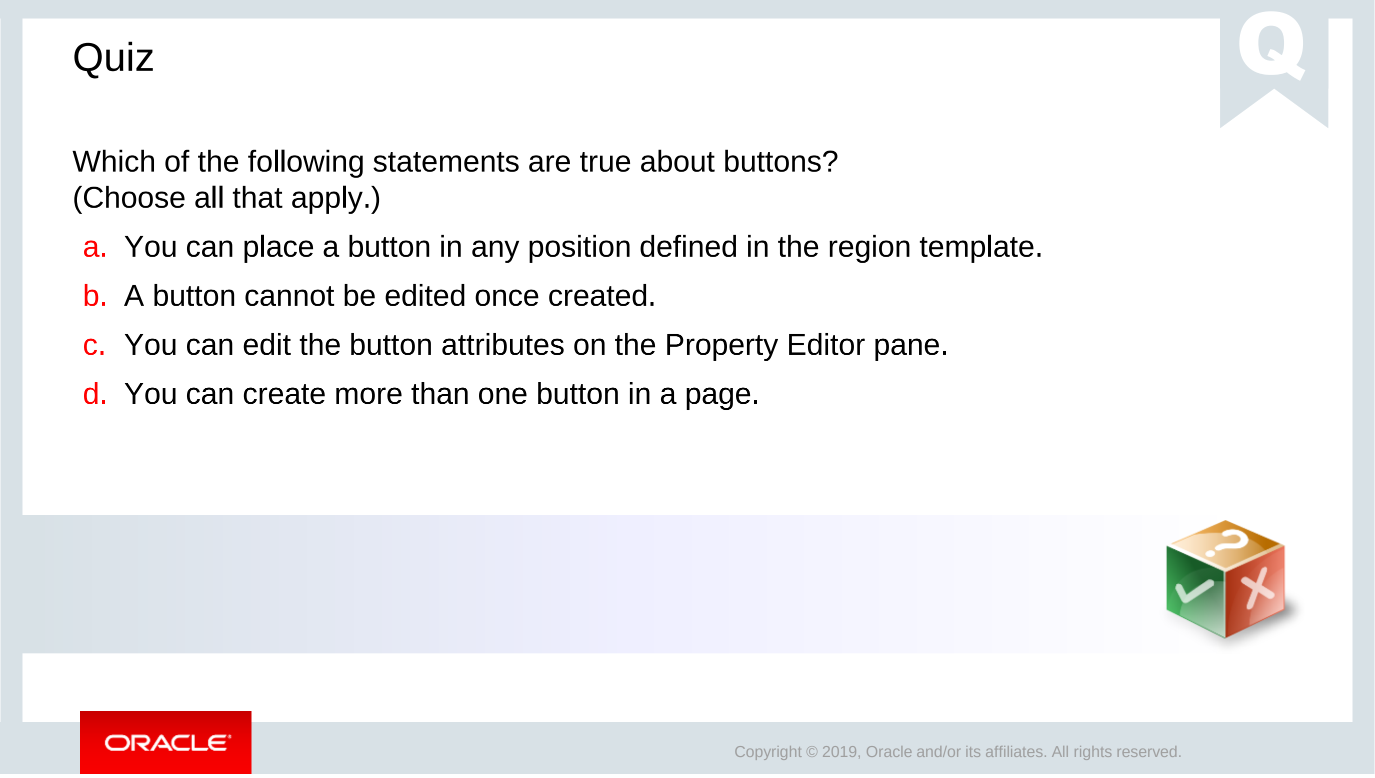
Type: URL

URL: www.oracle.com

1
2
Layout
Appearance
3
Cancel
Clear
OK
4
Project Tracking System
Form on Employees
Learn More
First Name
Last Name
Email
5
Would you like to visit an Oracle country site closer
Visit Oracle.com India →
No thanks, I'll stay here
See this page for different country
Menu
ORACLE
Platform as a Service (PaaS) for Dummies
Modernize your business in the cloud while leveraging artificial intelligence (AI)

Let's try modifying the button with a redirect to a URL. To edit a button, navigate to the Page Definition and perform the following steps:

1. In the Rendering pane, select the button name (*Learn More*) that you created a couple of slides back.
2. In the Property Editor pane, select the following:
   - **Behavior**: **Action**: Select **Redirect to URL**
   - **Behavior**: **Target**: Click **No Link Defined**
   - Enter the URL in the text area. In the slide example, the URL that is entered is `http://www.oracle.com` and click **OK**.
3. Save and run the page to check if redirect works when the button is clicked. You will see that it redirects to the Oracle website.



Quiz

Which of the following statements are true about buttons?
(Choose all that apply.)

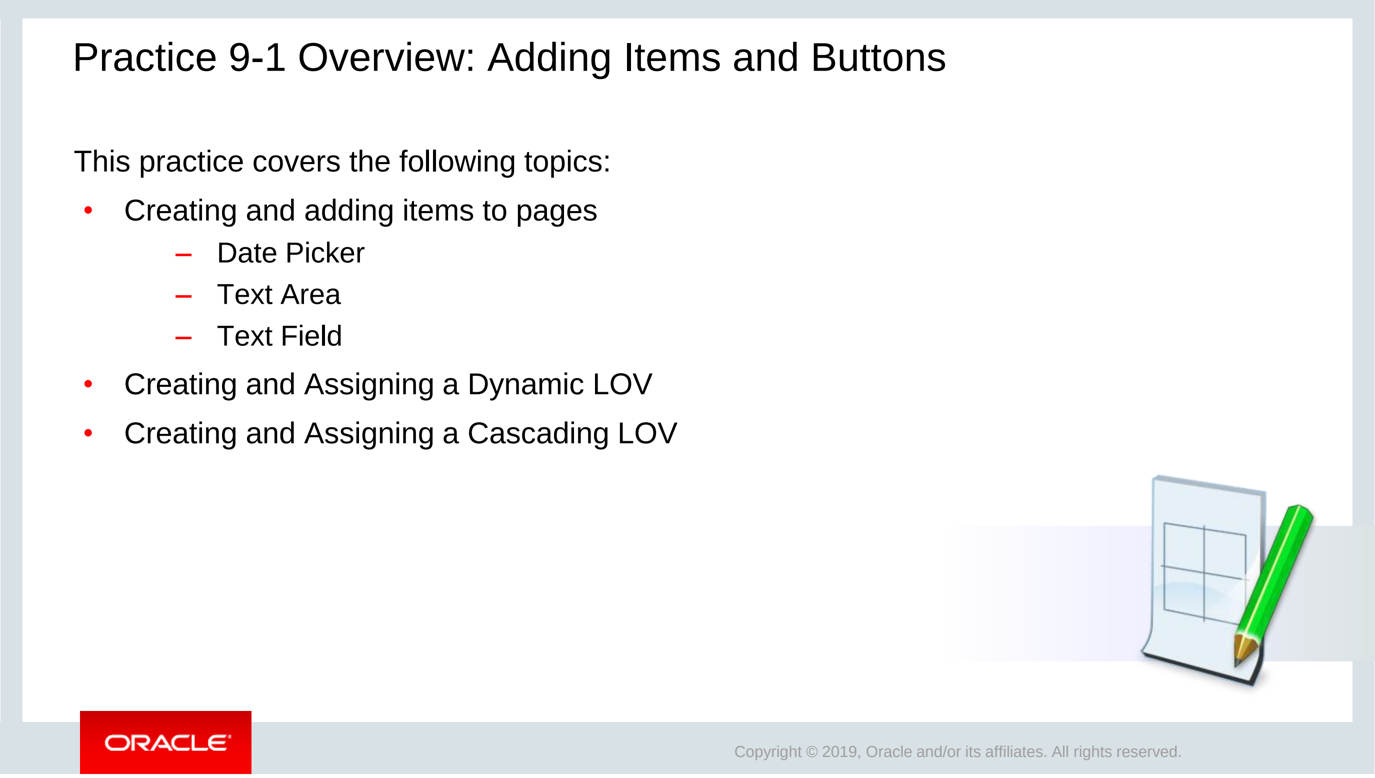
a. You can place a button in any position defined in the region template.

b. A button cannot be edited once created.

c. You can edit the button attributes on the Property Editor pane.

d. You can create more than one button in a page.

Answer: a, c, d



Practice 9-1 Overview: Adding Items and Buttons

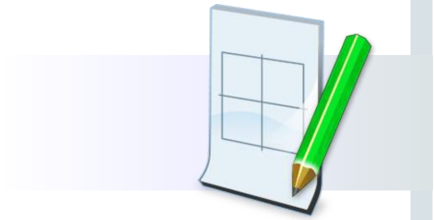
This practice covers the following topics:

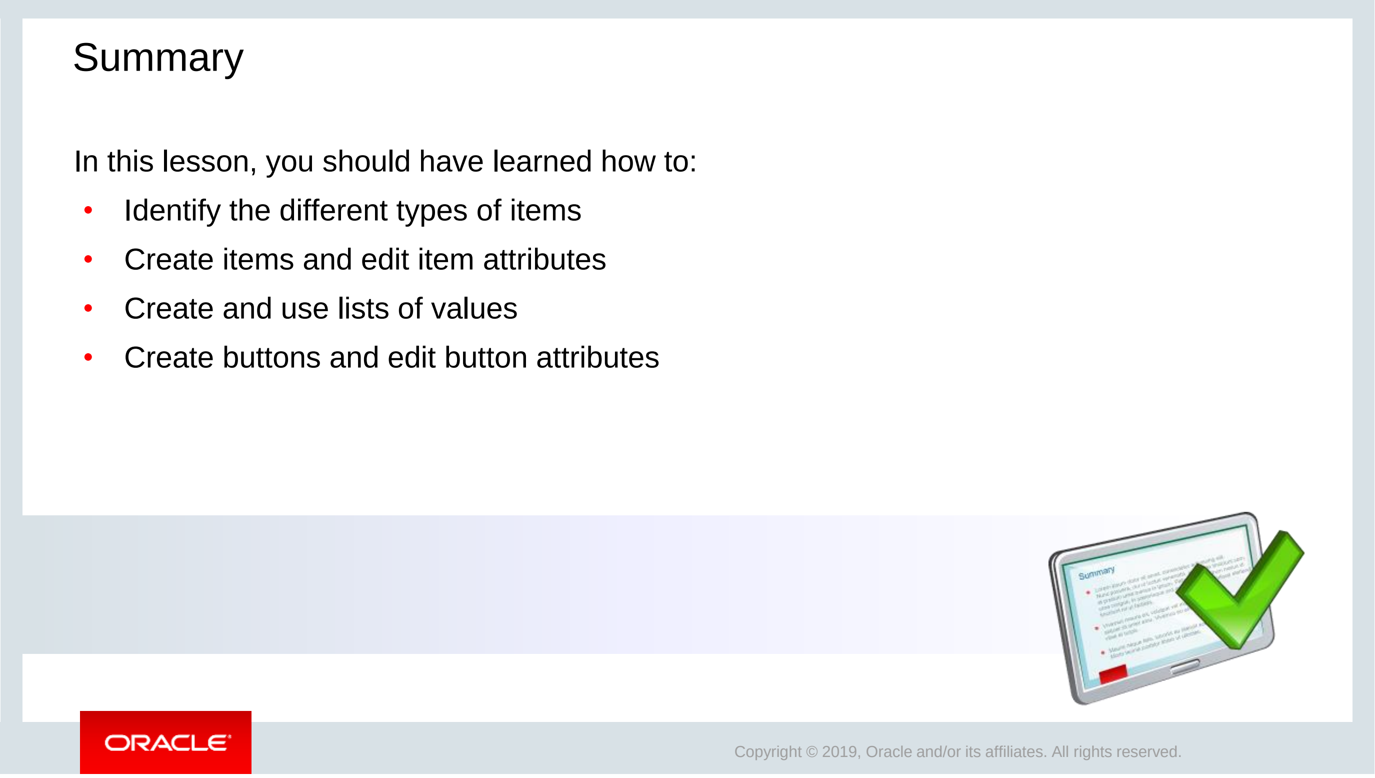
- Creating and adding items to pages
  - Date Picker
  - Text Area
  - Text Field
- Creating and Assigning a Dynamic LOV
- Creating and Assigning a Cascading LOV

Practice 9-2 Overview: Manipulating Items

This practice covers the following topics:

- Creating and adding Submit and Cancel buttons to the page
- Editing item and button attributes





Summary

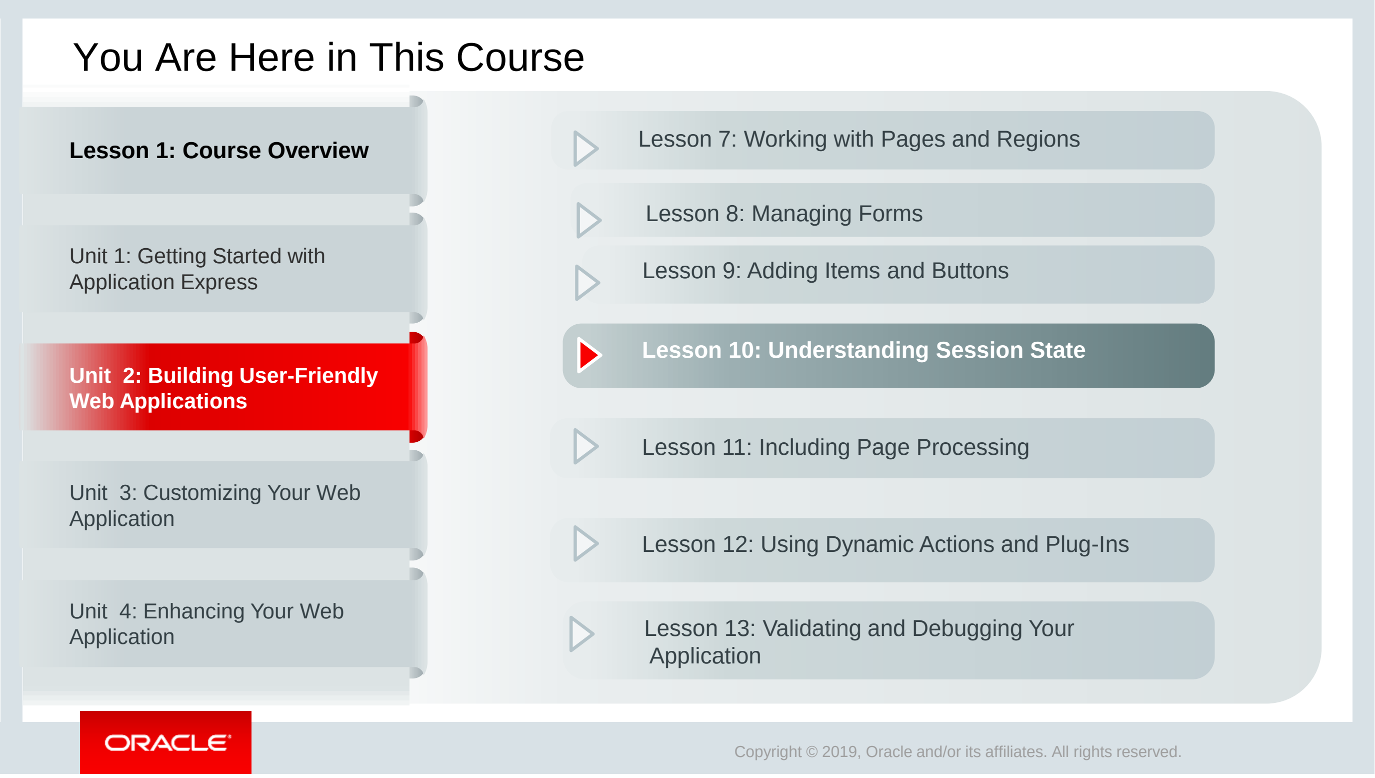
In this lesson, you should have learned how to:

- Identify the different types of items
- Create items and edit item attributes
- Create and use lists of values
- Create buttons and edit button attributes

In this lesson, you learned about items and buttons. You learned how to create items and buttons, as well as how to edit their attributes.

10

Understanding Session State



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

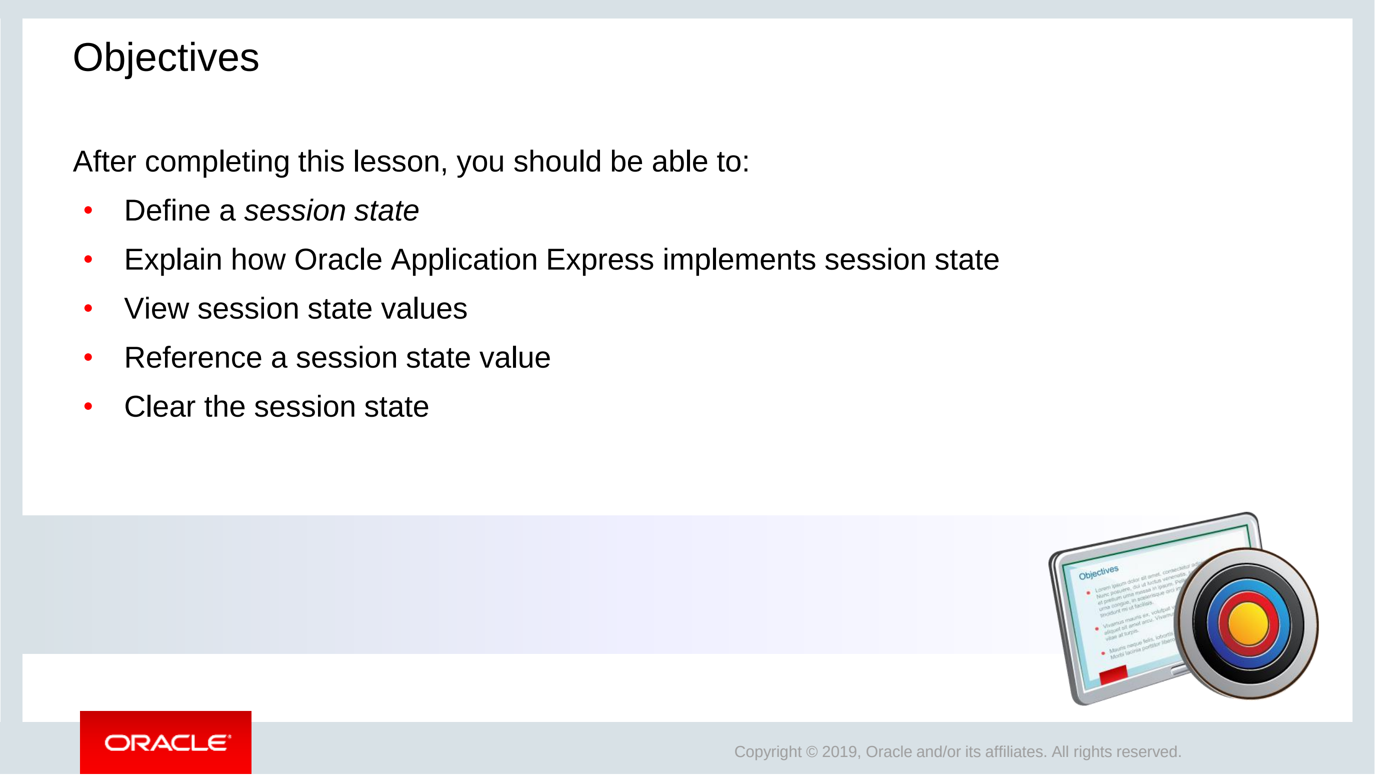
Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- Lesson 7: Working with Pages and Regions
- Lesson 8: Managing Forms
- Lesson 9: Adding Items and Buttons
- **Lesson 10: Understanding Session State**
- Lesson 11: Including Page Processing
- Lesson 12: Using Dynamic Actions and Plug-Ins
- Lesson 13: Validating and Debugging Your Application

This slide is a graphical depiction of the course, particularly highlighting Unit 2 - Lesson 10, which is dealt with in these slides.

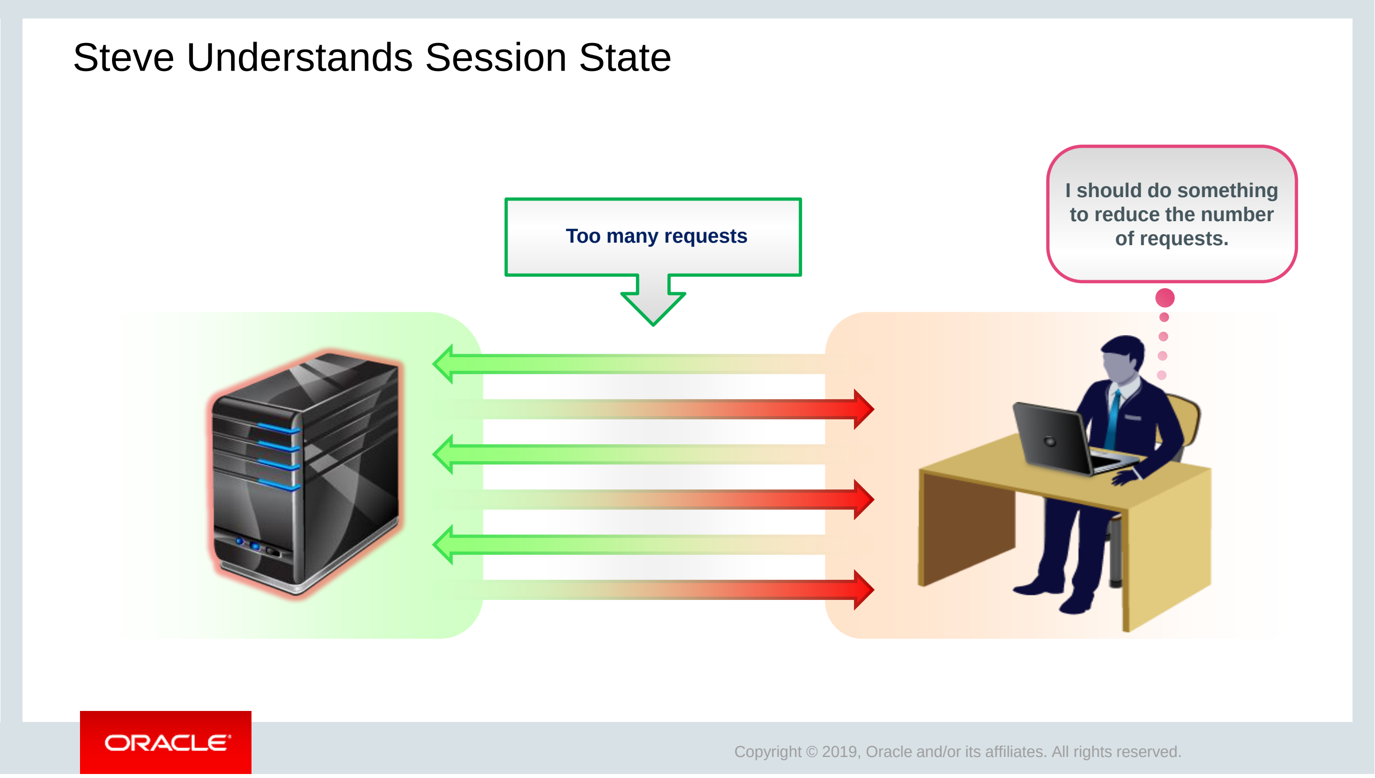


Objectives

After completing this lesson, you should be able to:

- Define a *session state*
- Explain how Oracle Application Express implements session state
- View session state values
- Reference a session state value
- Clear the session state

This lesson shows you how Oracle Application Express manages the session state of an application.



Steve Understands Session State

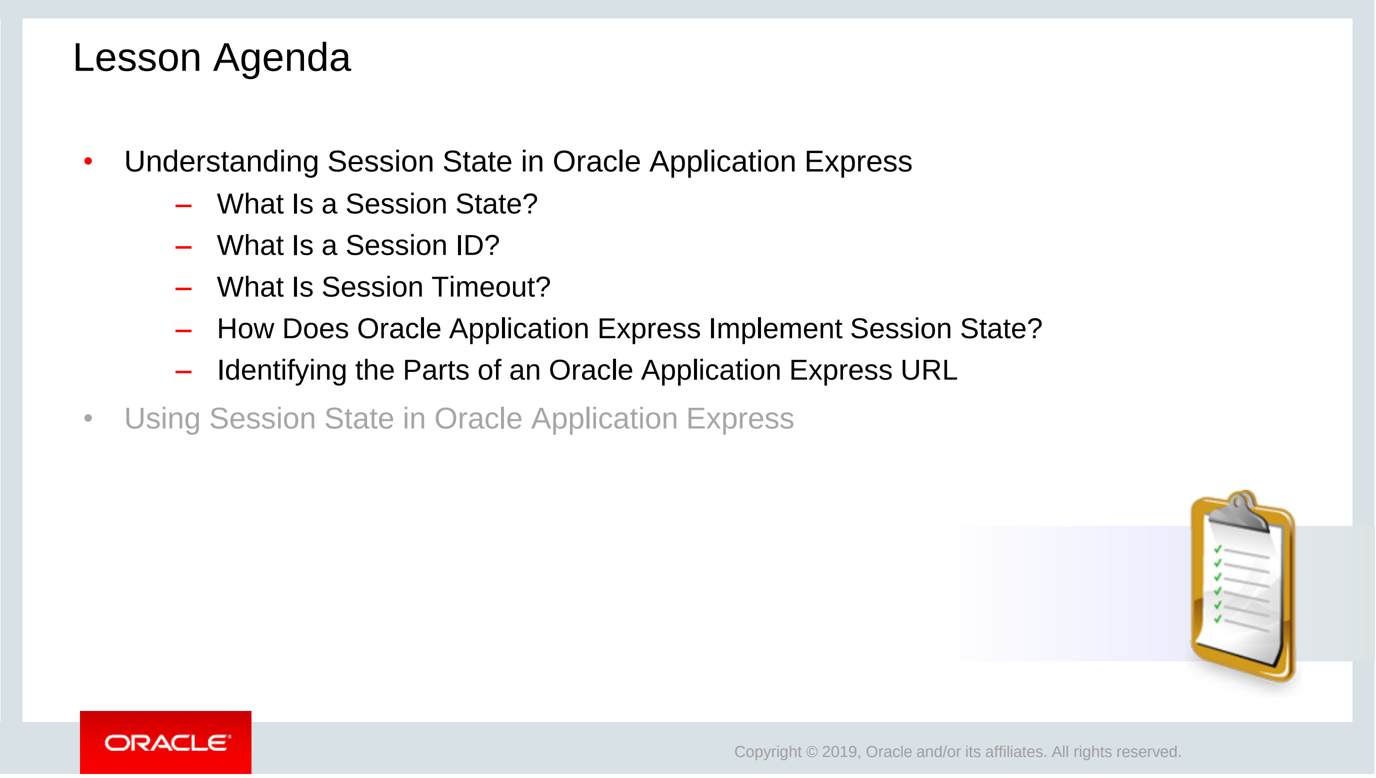
Too many requests

I should do something to reduce the number of requests.

Steve has the forms, reports, interfaces, and pages for the *Project Tracking System (PTS)* application ready and functional. At this point, each time any data is fetched from the server, the application sends a request to the server, thereby increasing the number of requests.

He is now looking for options in Oracle Application Express that can help him in maintaining the application state information so that the number of requests is reduced and values can be fetched from the session. In this lesson, he once more starts using the *Sample Database Application* as it is a fully functional and editable application. He wants to try these features once on this application before he applies it into his *PTS* application.

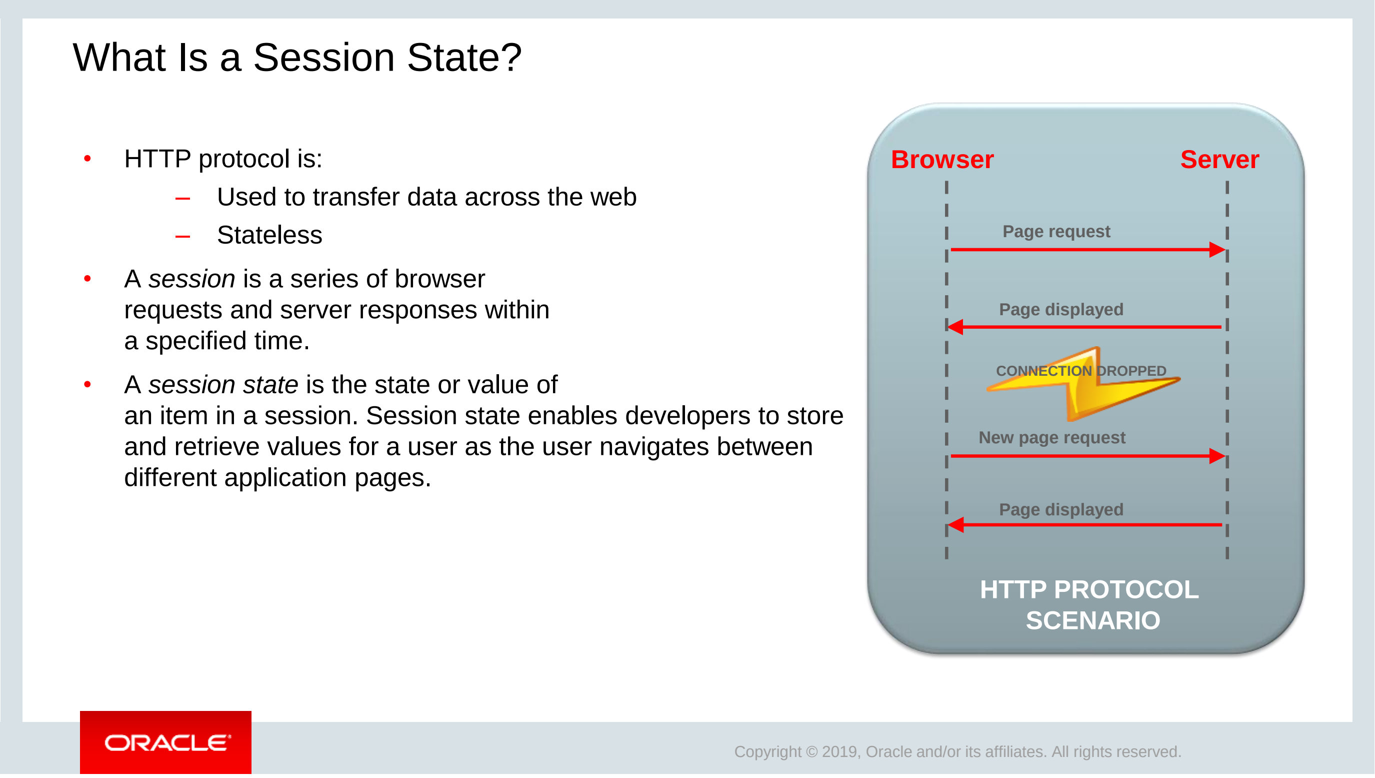
However, he switches to the *PTS* application while he works on clearing a user's session state (cache), in the later part of this lesson.



Lesson Agenda

- Understanding Session State in Oracle Application Express
  - What Is a Session State?
  - What Is a Session ID?
  - What Is Session Timeout?
  - How Does Oracle Application Express Implement Session State?
  - Identifying the Parts of an Oracle Application Express URL
- Using Session State in Oracle Application Express

ORACLE



What Is a Session State?

- HTTP protocol is:
  - Used to transfer data across the web
  - Stateless
- A *session* is a series of browser requests and server responses within a specified time.
- A *session state* is the state or value of an item in a session. Session state enables developers to store and retrieve values for a user as the user navigates between different application pages.

To understand what a session state is, you must first understand what HTTP is and how it works.

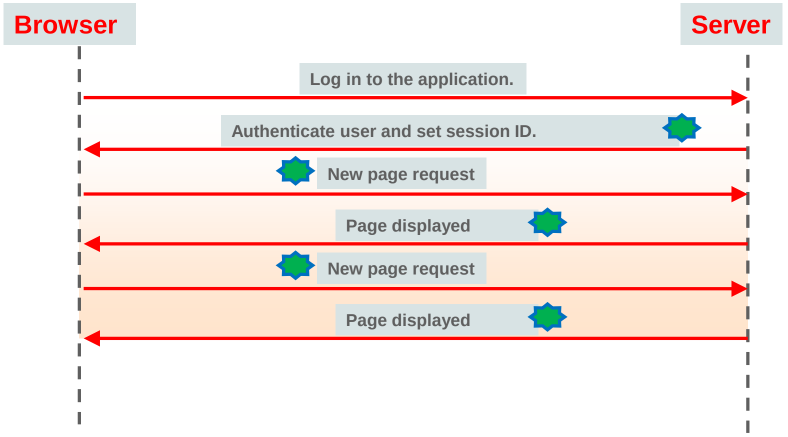
HTTP is the protocol that is used to transfer data across the web. HTTP is a stateless protocol. It means that each page request from a browser is treated as an independent request by the server. There is no memory or saved state between the requests.

In a web application scenario, such as an online shopping application, it is essential to maintain application state information. For example, a user fills out a web form for ordering products, then adds the items to be purchased, and finally submits the form. In this scenario, it is necessary to store the list of items in the shopping cart and then present this list when required, such as when confirming the order. In addition, the user information must also be retrievable when necessary. To access the values that are entered on one page from a different page, some sort of management is required.

A series of requests that originate from the same user by using the same web browser to a web server is called a *session*. The value of the page item during the session length is called the *session state* of the item.

Session ID

A session ID is a unique identifier that is assigned to each new session in an application.



Browser

Server

Log in to the application.

Authenticate user and set session ID.

New page request

Page displayed

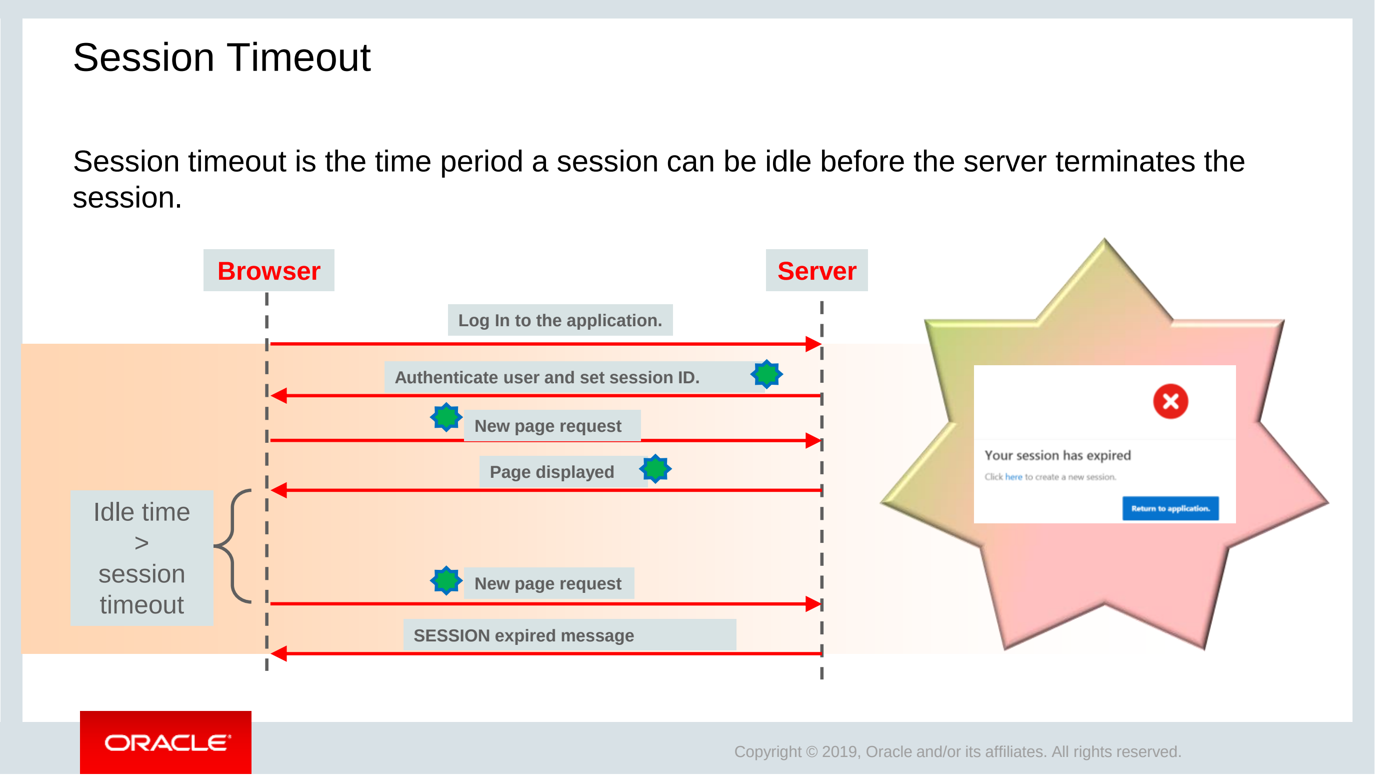
New page request

Page displayed

In the graphic in the slide, a user logs in to an application. The server authenticates the user and starts a new session. A *session ID* (depicted by a green symbol) is assigned to the session. Each time the browser makes a request to the server, the *session ID* is also sent to the server. The server uses this *session ID* to identify the user and maintain the session state for the user.

To manage sessions and to store session state information, each session should be uniquely identifiable by the server and the browser. This is done by using *session IDs*. A *session ID* is a unique identifier for each session created in an application. For each new session that is initiated by the browser, the server assigns a *session ID*. This *session ID* is associated with subsequent page requests, establishing a session.

There is also something called *session timeout*. You will learn about this in the next slide.



Session Timeout

Session timeout is the time period a session can be idle before the server terminates the session.

Your session has expired

Click here to create a new session.

Do you see the right-hand image above? What does it say? It says, “Your Session has expired,” and asks you to click a particular link to start a new session.

So, what is a *session timeout*? Session timeout is the time period set for an application session. If the user does not request a new page or refresh the current page within the time period, the server automatically terminates the session. By configuring the session timeout attributes, you can reduce your application's exposure. By setting the session and idle timeout, users are automatically logged out of their application after the specified timeout.

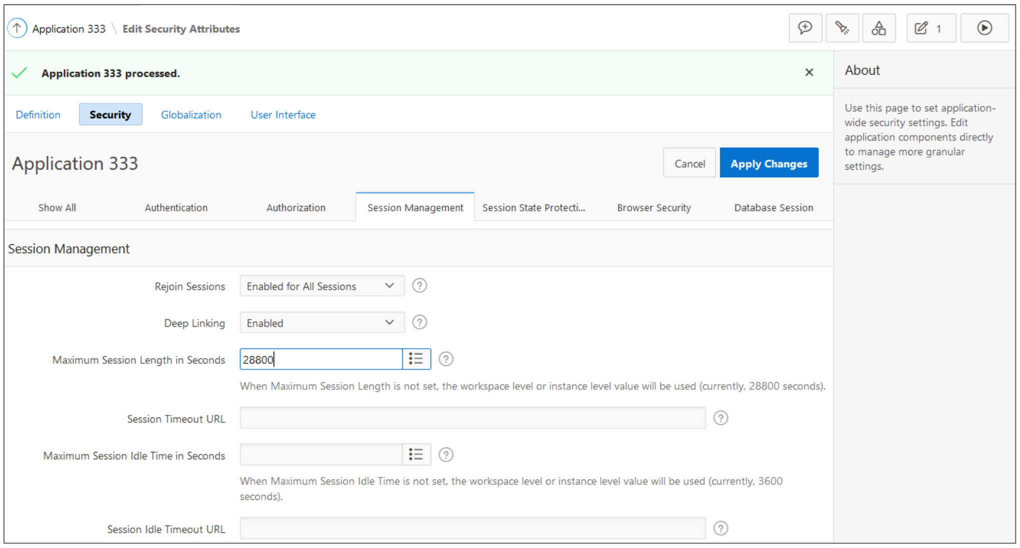
Session Timeout attributes include:

- ***Maximum Session Length in Seconds*****:** Enter a positive integer to control how many seconds a session exists and is used by this application.
- ***Session Timeout URL*****:** Enter an optional URL to redirect to when the maximum session lifetime has been exceeded.
- ***Maximum Session Idle Time in Seconds*****:** Enter a positive integer to control the seconds of inactivity or idle time for sessions used by this application. The idle time is the time between one page request and the next one.
- ***Idle Timeout URL*****:** Enter an optional URL to be redirected to when the maximum session idle time has been exceeded.

In the next slide, you can learn how to set the session timeout attributes in Oracle Application Express.

Setting Session Timeout

By configuring Session Timeout attributes, you can reduce your application's exposure.



Application 333 processed.

Use this page to set application-wide security settings. Edit application components directly to manage more granular settings.

Rejoin Sessions: Enabled for All Sessions

Deep Linking: Enabled

Maximum Session Length in Seconds: 28800

When Maximum Session Length is not set, the workspace level or instance level value will be used (currently, 28800 seconds).

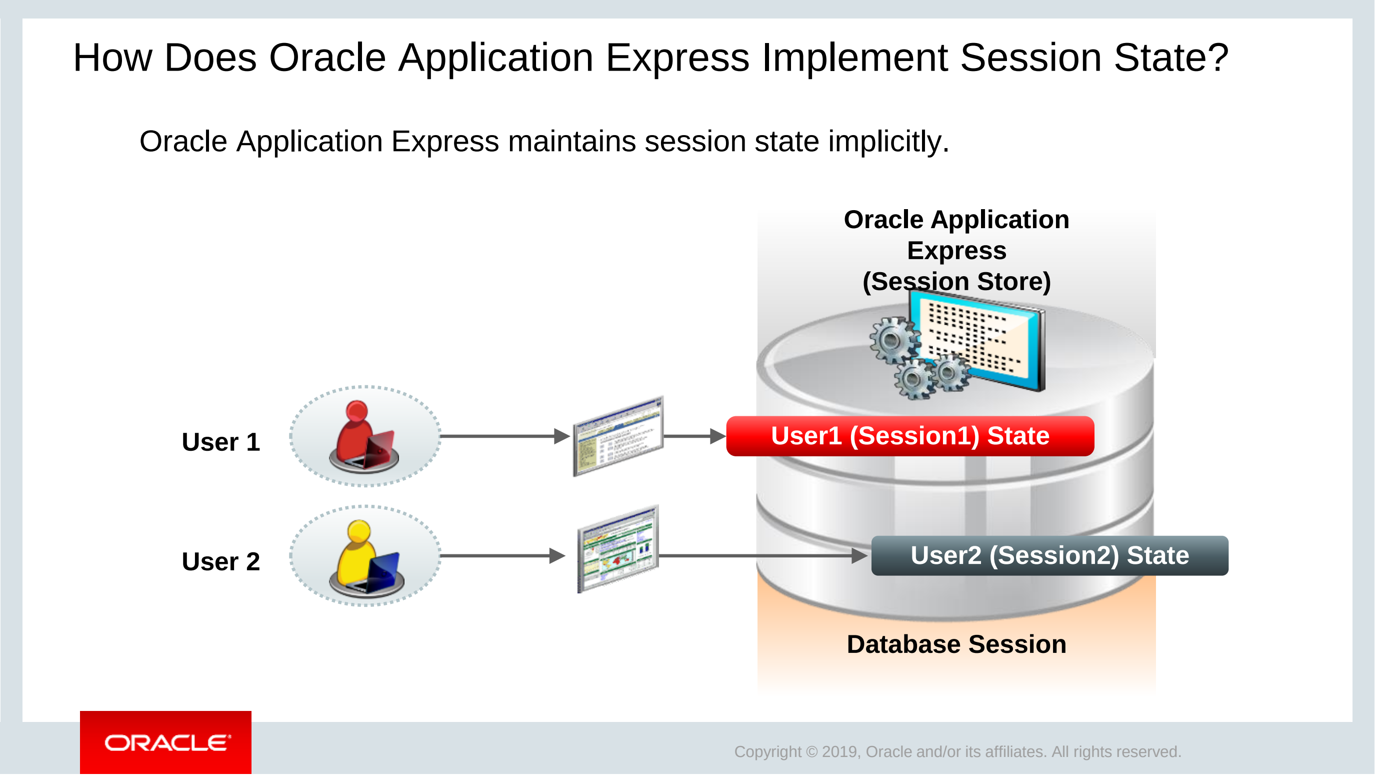
Session Timeout URL:

Maximum Session Idle Time in Seconds:

When Maximum Session Idle Time is not set, the workspace level or instance level value will be used (currently, 3600 seconds).

Session Idle Timeout URL:

You learned about the various session timeout attributes in the previous slide. In Oracle Application Express, you can declaratively specify session timeouts for maximum idle time and maximum session duration. To set the session timeout for an application, click the **Edit Application Properties** button on the application home page. Click the **Security** tab and then the **Session Management** tab.



How Does Oracle Application Express Implement Session State?

Oracle Application Express maintains session state implicitly.

Oracle Application Express (Session Store)

User 1

User1 (Session1) State

User 2

User2 (Session2) State

Database Session

ORACLE

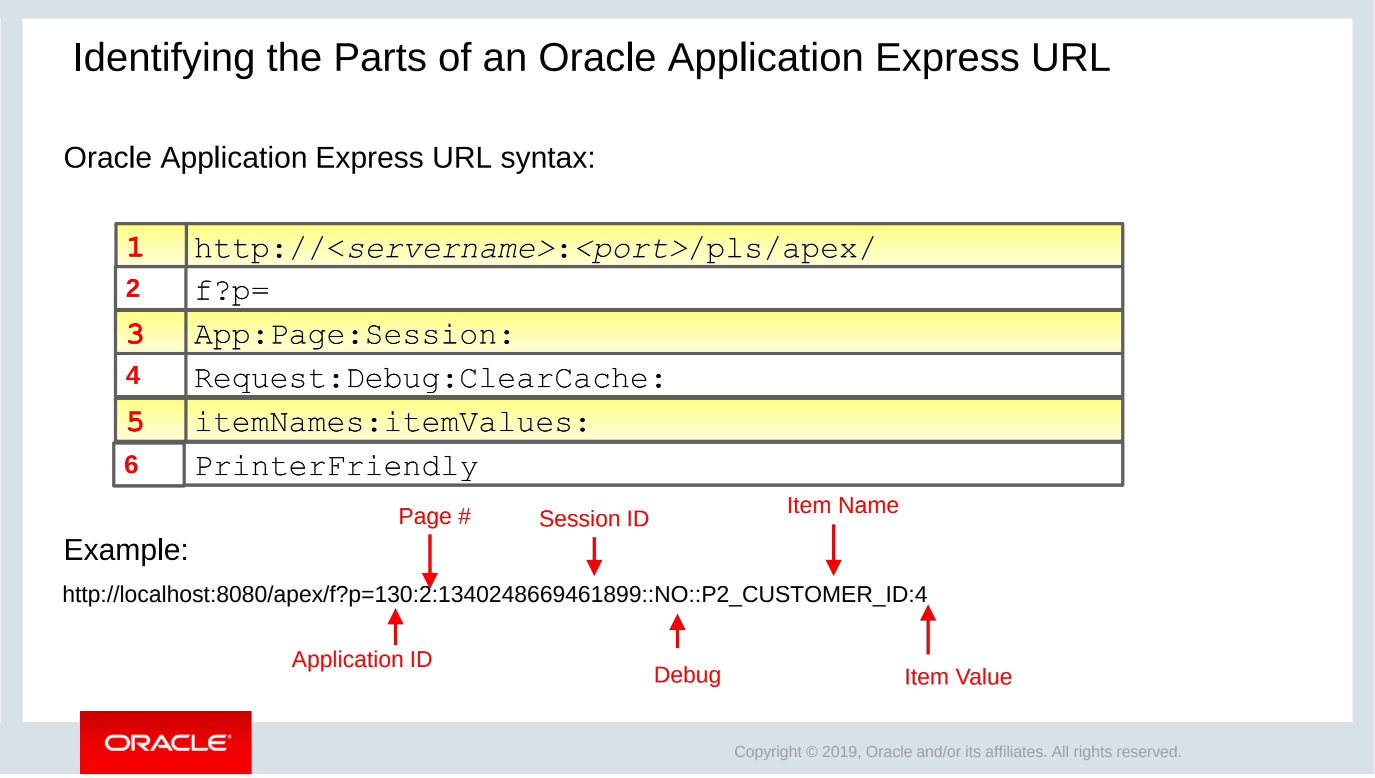
Now, you must be wondering how Oracle Application Express implements *session state*.

In Oracle Application Express, you do not need to write code to manage and maintain sessions or session state. *Session state* is maintained transparently, and you can easily access session state values and manipulate them, if required.

Oracle Application Express uses cookies to store session state. If you turn off cookies in your browser, Application Express applications will not work properly. The cookies hold information about the application, page, and so on. If developers run multiple instances that use the same browser on one PC when they build applications, then the different browser instances interfere with each other. When switching between the two different browser screens, the tool will exhibit strange behavior, including unexpected errors. This can be avoided by developing applications by using different browsers (such as Internet Explorer and Mozilla Firefox) because each browser tool uses its own cookies.

Each time users log in to an application, Oracle Application Express assigns a unique session identifier, which is associated with users until they log out of the application. This session ID is used by the Oracle Application Express engine to store and retrieve the application's working set of data before and after each page view. This is done by comparing the session ID with the session cookie and the session record in the database. The session cookie and the session record safeguard the integrity of the session ID and the authentication status of the user.

Multiple sessions can exist in the database at the same time, because Oracle Application Express treats each session independently. The session information persists in the database until it is purged. Therefore, as long as the client's session has not expired, a user can continue running the application long after having first launched it.



Identifying the Parts of an Oracle Application Express URL

Oracle Application Express URL syntax:

1	`http://<servername>:<port>/pls/apex/`
2	`f?p=`
3	`App:Page:Session:`
4	`Request:Debug:ClearCache:`
5	`itemNames:itemValues:`
6	`PrinterFriendly`

Example:

http://localhost:8080/apex/f?p=130:2:1340248669461899::NO::P2_CUSTOMER_ID:4

Page #

Session ID

Item Name

Application ID

Debug

Item Value

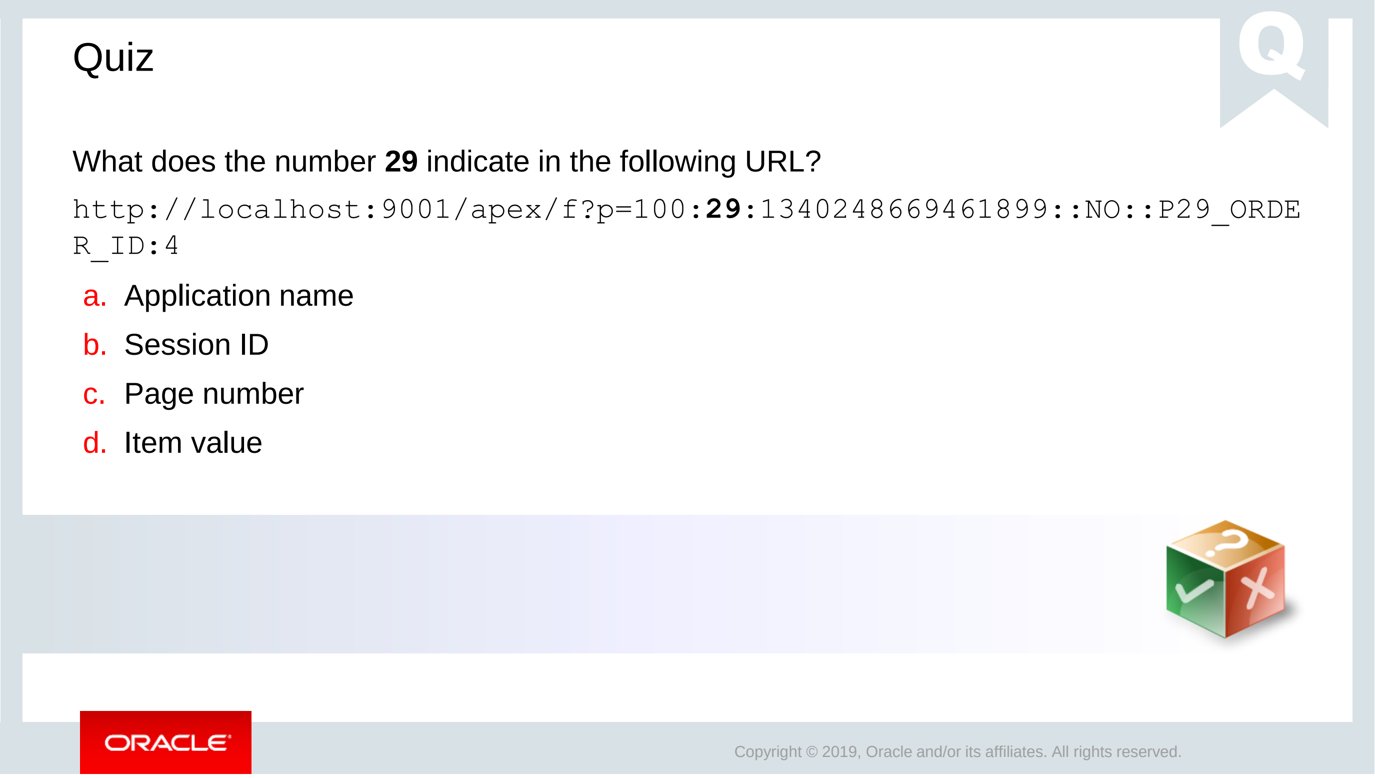
Let’s see now what each part of an Oracle Application Express URL stands for. The slide shows the syntax of a complete URL for an application developed by using Oracle Application Express.

1. The URL starts with the address of the Oracle Application Express instance. `pls` in the URL indicates that you are using Oracle HTTP Server with *`mod_plsql`*. If you are using Application Express Listener or Embedded PL/SQL Gateway, `pls` is omitted.
2. **`f`** is the procedure that is called and **`p`** is the rest of the URL that is passed as parameters to the procedure.
3. App: Page: Session:
   - `App` is the application ID or alias of the application that you want to access.
   - `Page` is the page number or alias of the page that you want to access.
   - `Session` is the identifier for the session assigned by Oracle Application Express when you log in to an application.
4. Request: Debug: ClearCache:
   - `Request` is set to the request attribute value of a button when it is clicked. For example, if you click a button called CREATE, CREATE is passed as `request` in the URL.
   - `Debug` can be set to `YES` (uppercase) to switch on the debug mode for your application. Every other value turns the debugger off.
   - `ClearCache` is used to set the session state values to null. To clear a page/clear cache, specify the page number. To clear multiple pages, specify a comma-separated list of page numbers.

5. `itemNames` is a comma-separated list of item names and `itemValues` is a comma-separated list of item values. Item values cannot include colons, but can contain commas if enclosed with backslashes. To pass a comma in an item value, enclose the characters with backslashes (for example, `\123,45\`).
6. `PrinterFriendly` can be set to `YES` to render the page by using the printer-friendly page template.

For more details, see *Understanding URL Syntax* in Oracle Application Express documentation (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/understanding-url-syntax.html#GUID-F9B81EAF-D33F-401D-8349-3952DEDA5460)

In the next few slides, you will be learning how to view session state values, reference a session state value, and clear the session state before loading the page next time.



Quiz

What does the number **29** indicate in the following URL?

```
http://localhost:9001/apex/f?p=100:29:1340248669461899::NO::P29_ORDER_ID:4
```

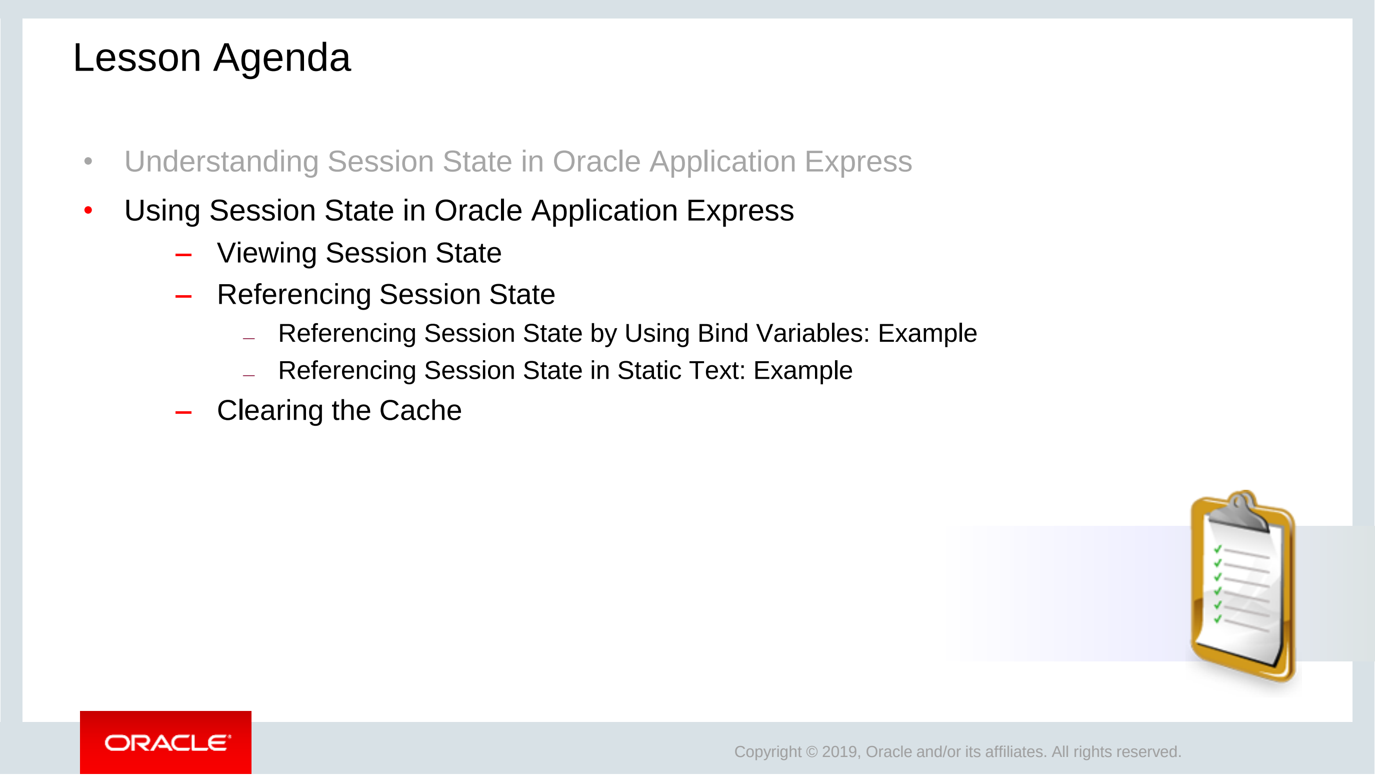
a. Application name

b. Session ID

c. Page number

d. Item value

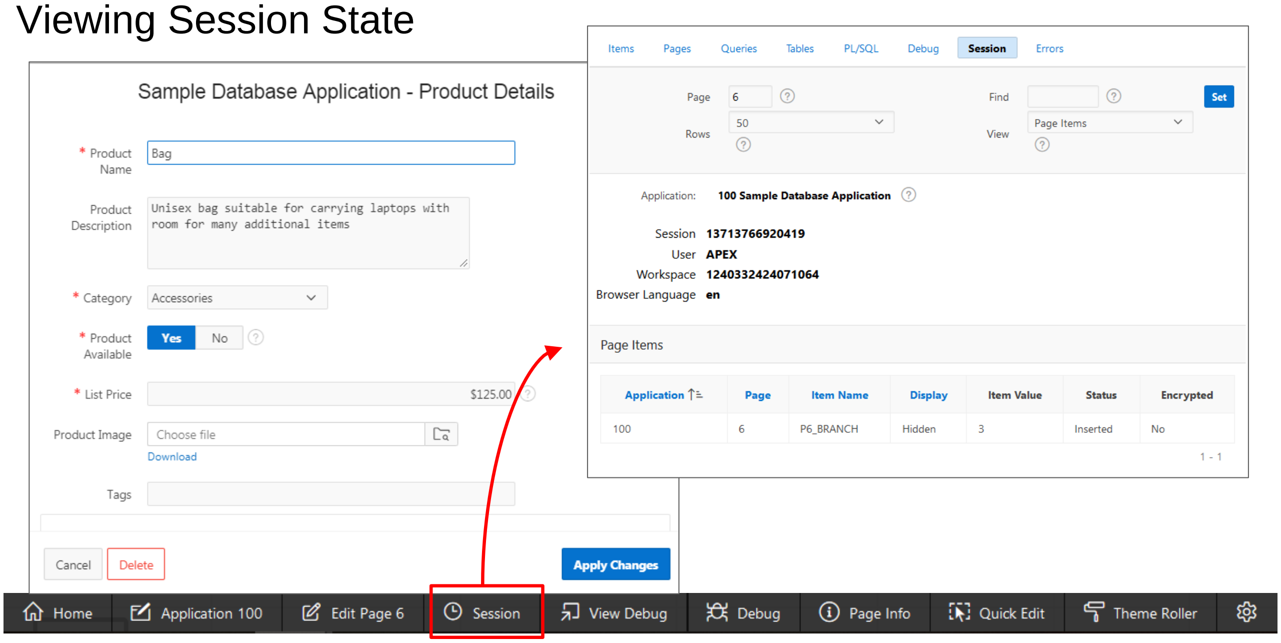
Answer: c



Lesson Agenda

- Understanding Session State in Oracle Application Express
- Using Session State in Oracle Application Express
  - Viewing Session State
  - Referencing Session State
    - Referencing Session State by Using Bind Variables: Example
    - Referencing Session State in Static Text: Example
  - Clearing the Cache

Viewing Session State



Sample Database Application - Product Details

* Product Name	Bag
Product Description	Unisex bag suitable for carrying laptops with room for many additional items
* Category	Accessories
* Product Available	Yes / No
* List Price	$125.00
Product Image	Choose file Download
Tags	

Page: 6 Rows: 50 Find: View: Page Items

Application: 100 Sample Database Application

Session 13713766920419
User APEX
Workspace 1240332424071064
Browser Language en

Page Items

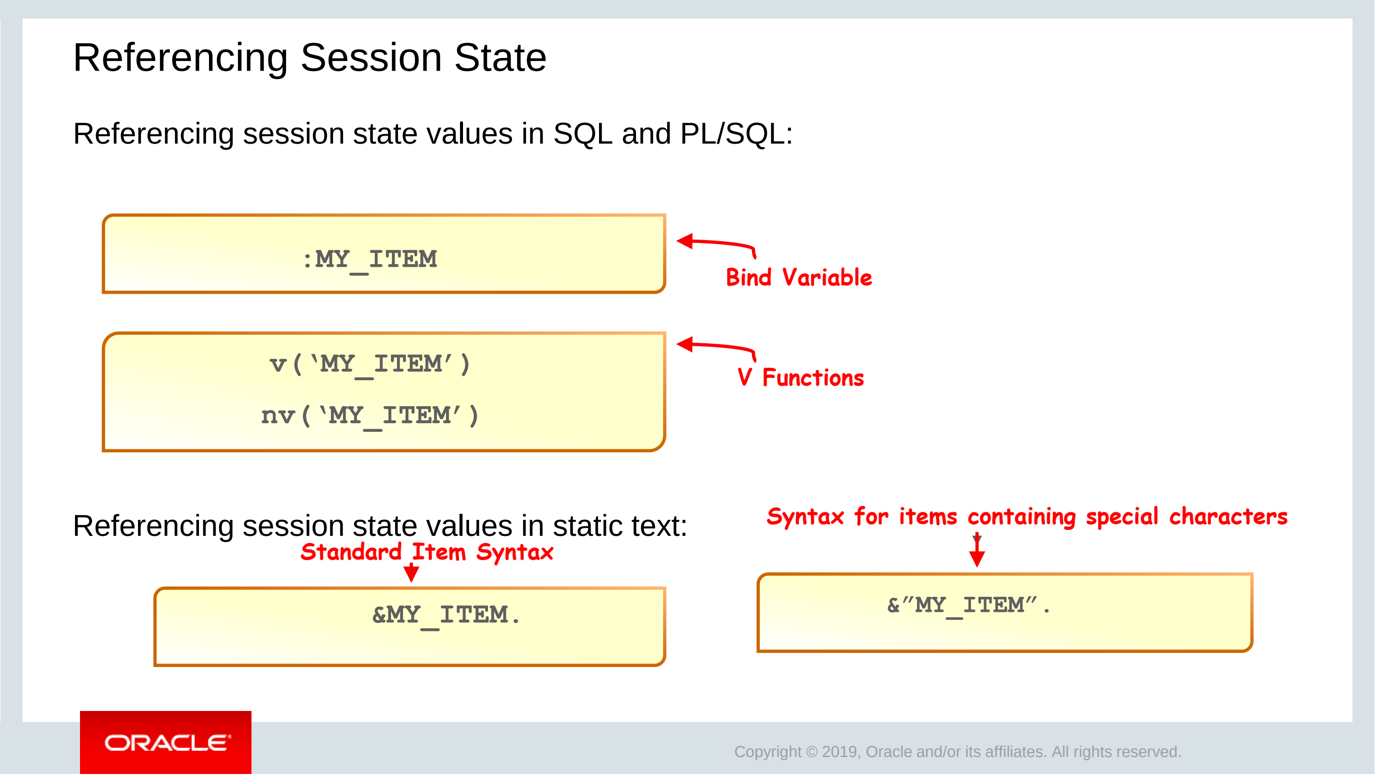
Application	Page	Item Name	Display	Item Value	Status	Encrypted
100	6	P6_BRANCH	Hidden	3	Inserted	No

1 - 1

You learned about *session state* a few slides back. Now, let’s see how you can view the *session state* of a particular page (for example, *Product Details* page in the *Sample Database Application*). To view the session state for a page, click the **Session** button on the Developer toolbar. The Session State page opens in a new window and provides information about a page, such as:

- Session ID, current user, workspace ID, and the browser language
- The attributes of the page, such as the item name, how the item is displayed, the state or session ID, and the status. The status column indicates the status of the session state. The values include I (Inserted), U (Updated), and R (Reset).
- The application items that do not reside on a page. The application items are session state variables without the associated user interface properties. Application items are not used for display, but used as global variables to the application.

When you view a page for the first time, before making any changes and submitting the page, the state column on the session page is displayed as null. After you click a button and submit the page, when you view the session page, the state column displays the item values, and the status column shows that an insert operation has been performed.



Referencing Session State

Referencing session state values in SQL and PL/SQL:

`:MY_ITEM`

Bind Variable

`v('MY_ITEM')`

`nv('MY_ITEM')`

V Functions

Referencing session state values in static text:

Standard Item Syntax

`&MY_ITEM.`

Syntax for items containing special characters

`&"MY_ITEM".`

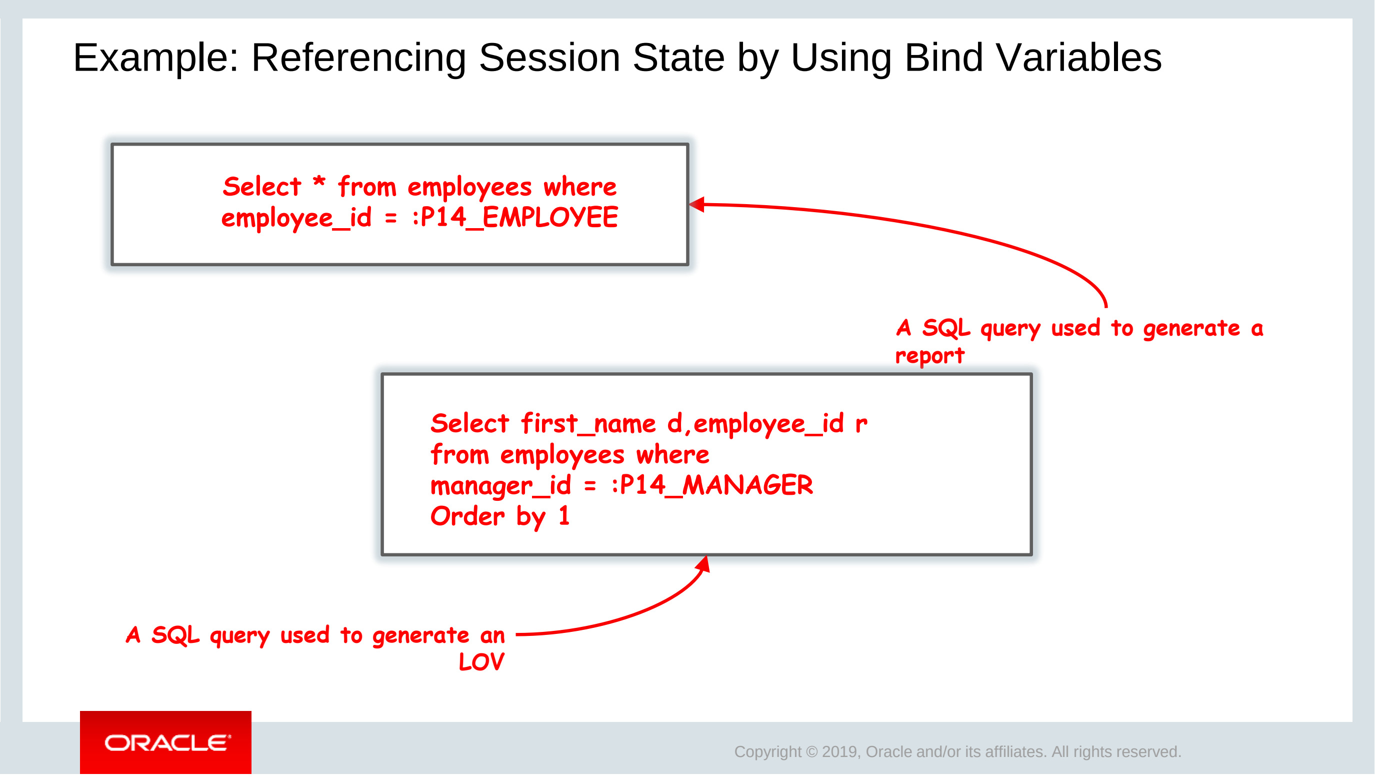
Steve understands that when building interactive, data-driven web applications, the ability to access and manage session state values is critical. However, in Oracle Application Express, session state is automatically managed for every page and easily referenced in static HTML or logic controls such as processes or validations. You will learn about processes and validations in a later chapter.

There are many scenarios where you can use bind variables to reference the session state value. For example, if you want to display a report and restrict the result based on some item values entered or selected by the user, you can use bind variables to reference the item's session state value in the `WHERE` clause of the SQL query. Similarly, if you want to display a list of values depending on user input, you can use a bind variable in the SQL query. Now, the supported syntax for referencing item values is:

- SQL and PL/SQL
  - Use the standard bind variable syntax for item names that are no longer than 30 characters. You can use this syntax for references within a SQL or PL/SQL query (for example, `:MY_ITEM`).
  - Use the `v` function to reference the item value [for example, `v('MY_ITEM')]` if the item name is longer than 30 characters or when you are coding a stored procedure.
  - Use the `nv` function to reference numeric items [for example, `nv('MY_NUMERIC_ITEM')].`

- Static text
    - Standard item syntax: Use `&item name` followed by a period “.” (for example, `&MY_ITEM.`).
    - Syntax for items containing special characters (to reference page items containing special, multibyte, or unicode characters): Wrap the page item name in double quotation marks (for example, `&"MY_ITEM”`).

In the next two slides, you see some examples of referencing session state values in static text and using bind variables.



Example: Referencing Session State by Using Bind Variables

```
Select * from employees where
employee_id = :P14_EMPLOYEE
```

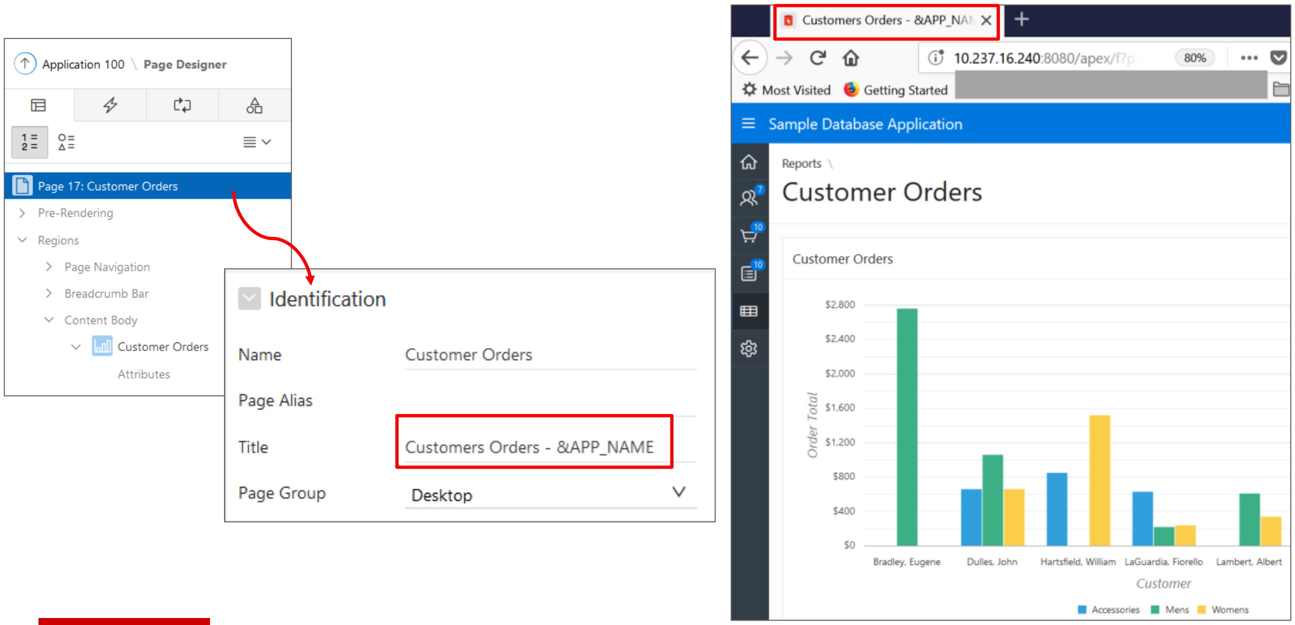
A SQL query used to generate a report

```
Select first_name d,employee_id r
from employees where
manager_id = :P14_MANAGER
Order by 1
```

A SQL query used to generate an LOV

In this slide, Steve uses two example SQL queries. The first one displays a report with a list of employees. The second one generates an LOV where you can select the employees reporting to a particular manager, via the manager ID.

Example: Referencing Session State in Static Text



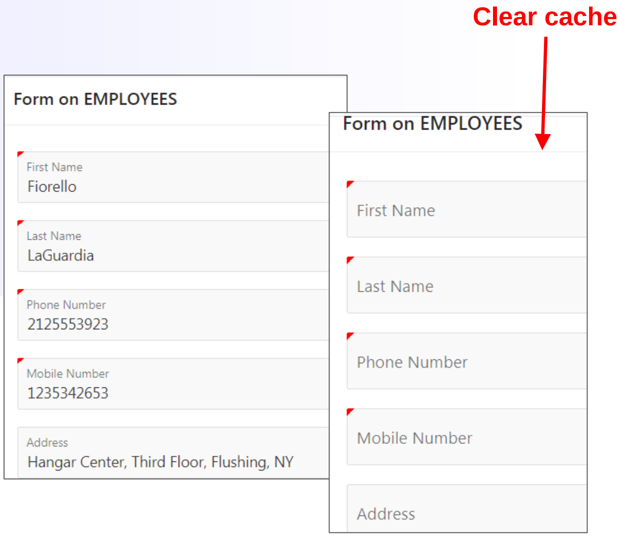
Identification

Name	Customer Orders
Page Alias	
Title	Customers Orders - &APP_NAME
Page Group	Desktop

A couple of slides back, you learned that Static Text is one of the supported syntax for referencing item values. In this example, Steve uses Static Text to display the name of the application appended to the **Customers Orders** Title (`Customers Orders – &APP_NAME)` of the page. Notice the title of the page (highlighted in red).

Clearing the Cache

1. In the Page Designer, select the relevant button in the Rendering pane.
2. Click the page number button in the Target field under Behavior.
3. In the Clear Cache field, enter the page number of the page for which you want to clear the cache and click OK.
4. Save and run the page.



Clear cache

Form on EMPLOYEES

First Name	Fiorello
Last Name	LaGuardia
Phone Number	2125553923
Mobile Number	1235342653
Address	Hangar Center, Third Floor, Flushing, NY

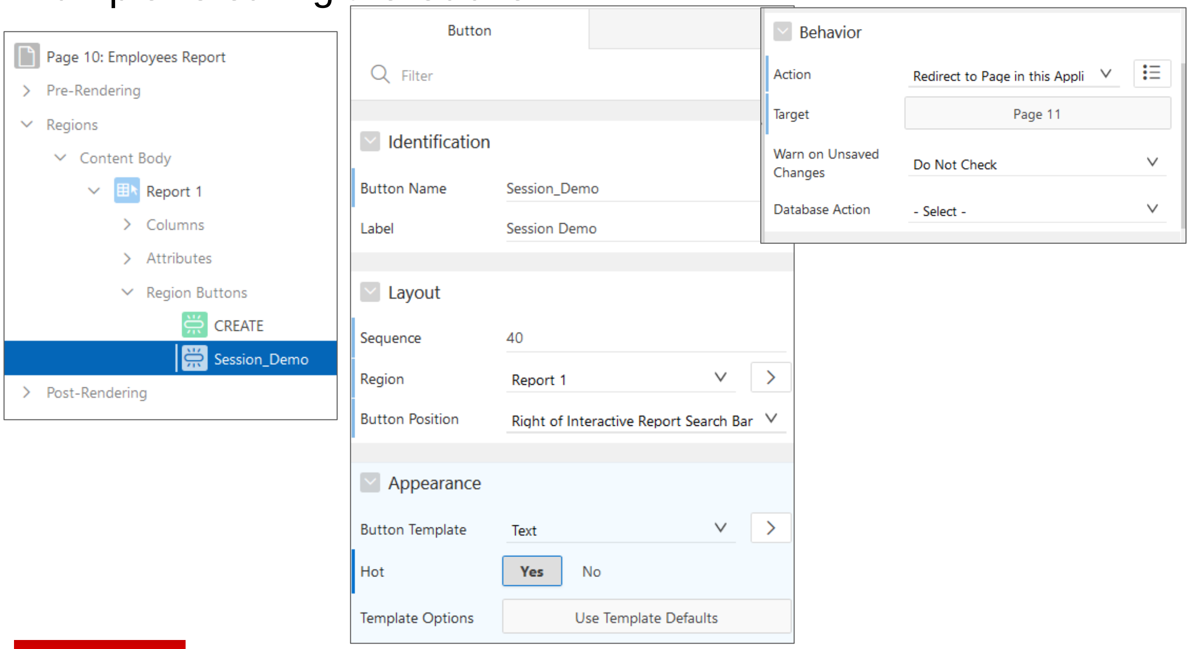
Form on EMPLOYEES

First Name	
Last Name	
Phone Number	
Mobile Number	
Address	

Now, the problem in the *PTS* application is that when a user clicks a form, for example, *Create Employees,* and sees the employee details, it remains in the cache and populates the same values when you again click it. Since there will be multiple managers along with Stella who would be using this, Steve wants to have a clear page every time a user opens up this form. In the next few slides, let’s see how Steve clears the user’s session state (or cache).

This slide provides an overview of the steps to clear the cache. Clearing a cached value resets the value to null. You can clear the cached value for specific items, all items on a page, all pages in an application, or the current user session.

Example: Clearing the Cache



- Page 10: Employees Report
  - Pre-Rendering
  - Regions
    - Content Body
      - Report 1
        - Columns
        - Attributes
        - Region Buttons
          - CREATE
          - Session_Demo
  - Post-Rendering

Identification

Button Name	Session_Demo
Label	Session Demo

Layout

Sequence	40
Region	Report 1
Button Position	Right of Interactive Report Search Bar

Appearance

Button Template	Text
Hot	Yes No
Template Options	Use Template Defaults

Behavior

Action	Redirect to Page in this Appli
Target	Page 11
Warn on Unsaved Changes	Do Not Check
Database Action	- Select -

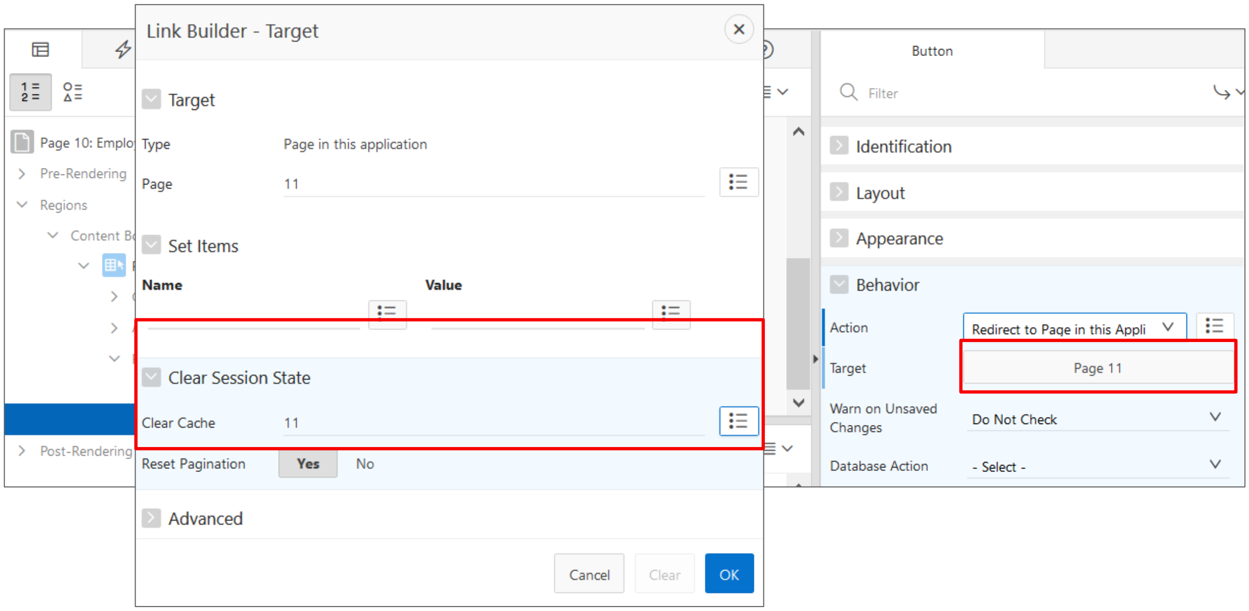
In the lesson on *Items and Buttons,* we discussed that you will be creating a button named *Session Demo*, which clears the cache for the items on a page and redirects to another page. Let's see how to do it in this slide.

First, Steve switches to his *PTS* application and adds a new button called `Session Demo`, which navigates to the *Create Employees* page when it is clicked.

1. Select the **Employees Report** page.
2. Right-click **Report 1** and select **Create Button**.
3. Select the newly created button. In the Button tab, enter the following values for the respective fields:
   - **Identification**: **Button Name**: Enter `Session_Demo`
   - **Layout**: **Button Position**: Select **Right of Interactive Report Search Bar**
   - **Appearance**: **Hot**: Select **Yes**
   - **Appearance**: **Template Options**: Select **Use Template Defaults**
   - **Behavior**: **Action**: Select **Redirect to Page in this Application**
   - **Behavior**: **Target**: Select **No Link Defined**
   - Click the *Up* Arrow near the **Page** field.
   - Select the **Page 11: Create Employees** page from the list. Click **OK**.

The *Session Demo* button is created in the *Employees Report* page. When you click this button, it navigates to the *Create Employees* page. As you learned in the previous slide, when a user clicks this button and sees the employee details, it remains in the cache and populates the same values when you again click it. But Steve wants to have a clear page every time a user clicks the *Session Demo* button. Let's see how he does it in the next slide.

Example: Clearing the Cache



Link Builder - Target

Target

Type: Page in this application

Page: 11

Set Items

Name	Value

Clear Session State

Clear Cache: 11

Reset Pagination: Yes No

Advanced

Action: Redirect to Page in this Appli

Target: Page 11

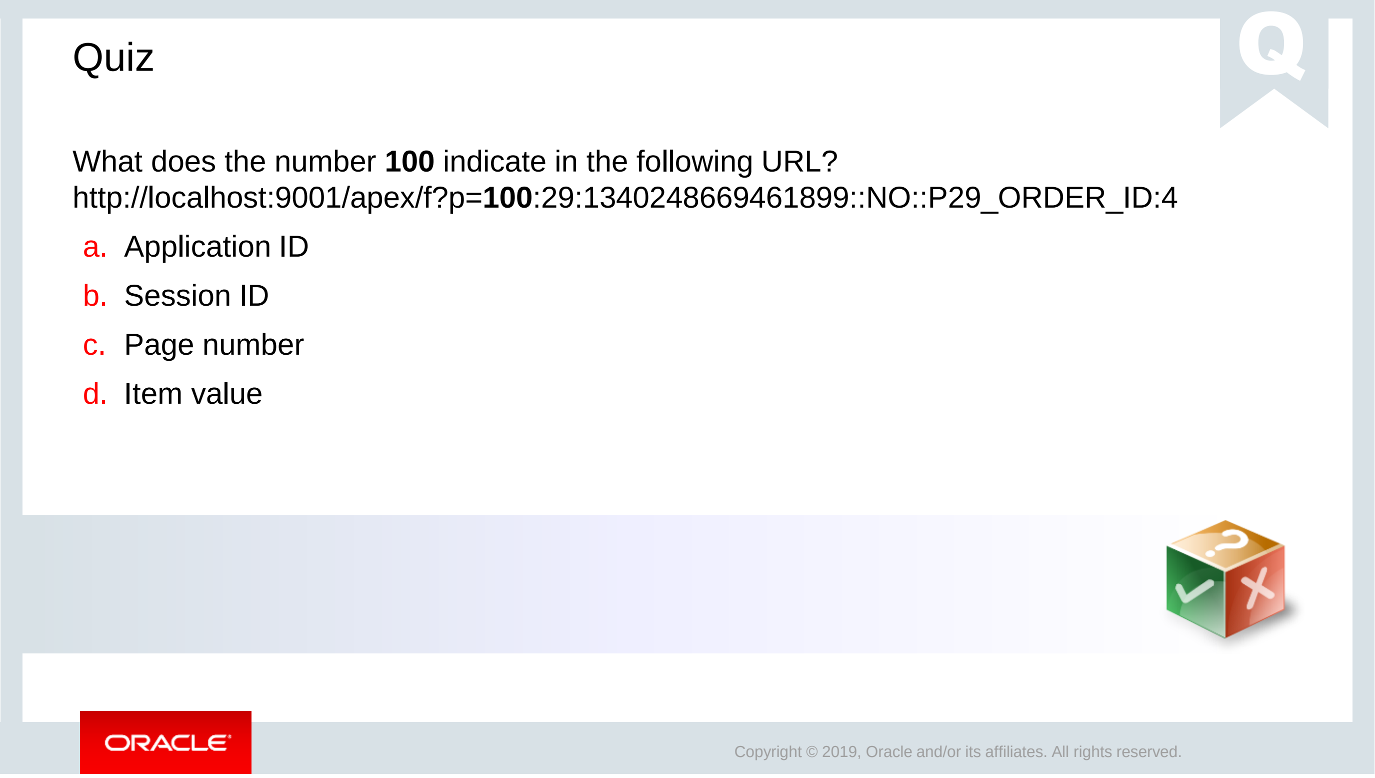
Warn on Unsaved Changes: Do Not Check

Database Action: - Select -

You can clear the page cache by using a built-in Oracle Application Express process. This slide shows an example of how Steve clears the values stored in the cache.

1. In the Rendering tab, select the **Session Demo** button.
2. Click the **Behavior: Target: Page 11: Create Employees**.
3. For **Clear Cache**, enter the page number of the page you want to clear the cache of (in this example, **Page 11: Create Employees**) and click **OK**.
4. Click the **Save and Run Page** icon. Click **Session Demo** again. Note that the values are no longer cached and the form is now empty.

Note: To clear the cache on multiple pages, you can enter multiple page numbers in the **Clear Cache** field. For example, if you enter 11,17,18, the cache of pages 11, 17, and 18 is cleared.

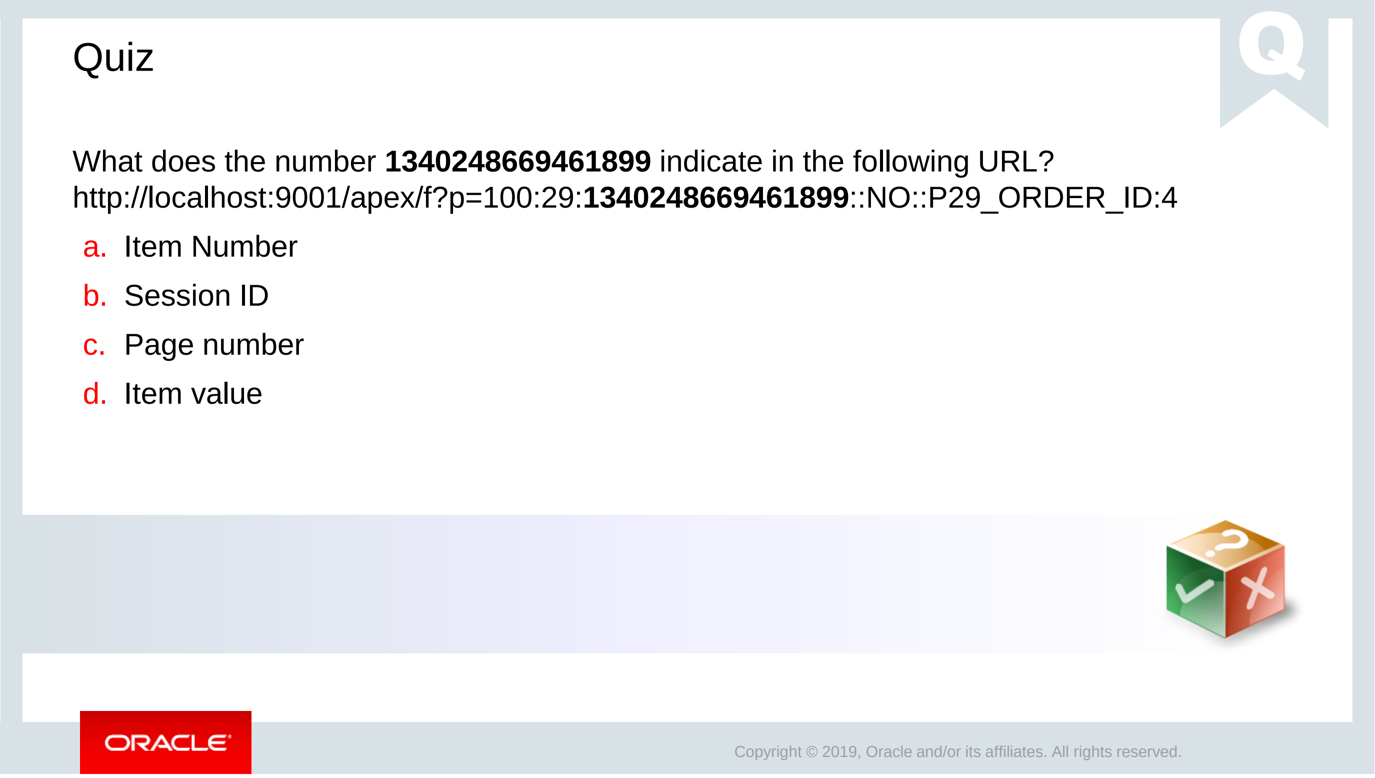


Quiz

What does the number **100** indicate in the following URL?
http://localhost:9001/apex/f?p=**100**:29:1340248669461899::NO::P29_ORDER_ID:4

a. Application ID
b. Session ID
c. Page number
d. Item value

Answer: a



Quiz

What does the number **1340248669461899** indicate in the following URL?
http://localhost:9001/apex/f?p=100:29:**1340248669461899**::NO::P29_ORDER_ID:4

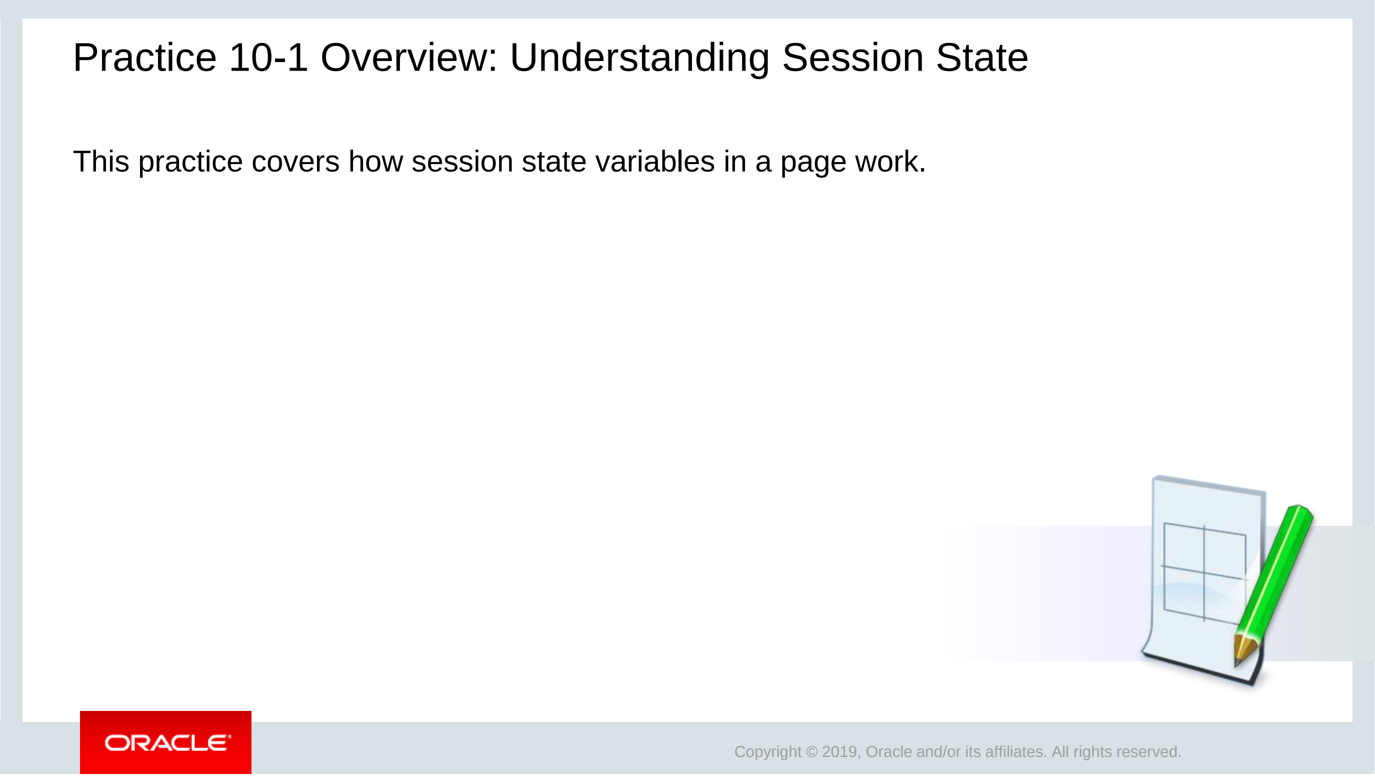
a. Item Number

b. Session ID

c. Page number

d. Item value

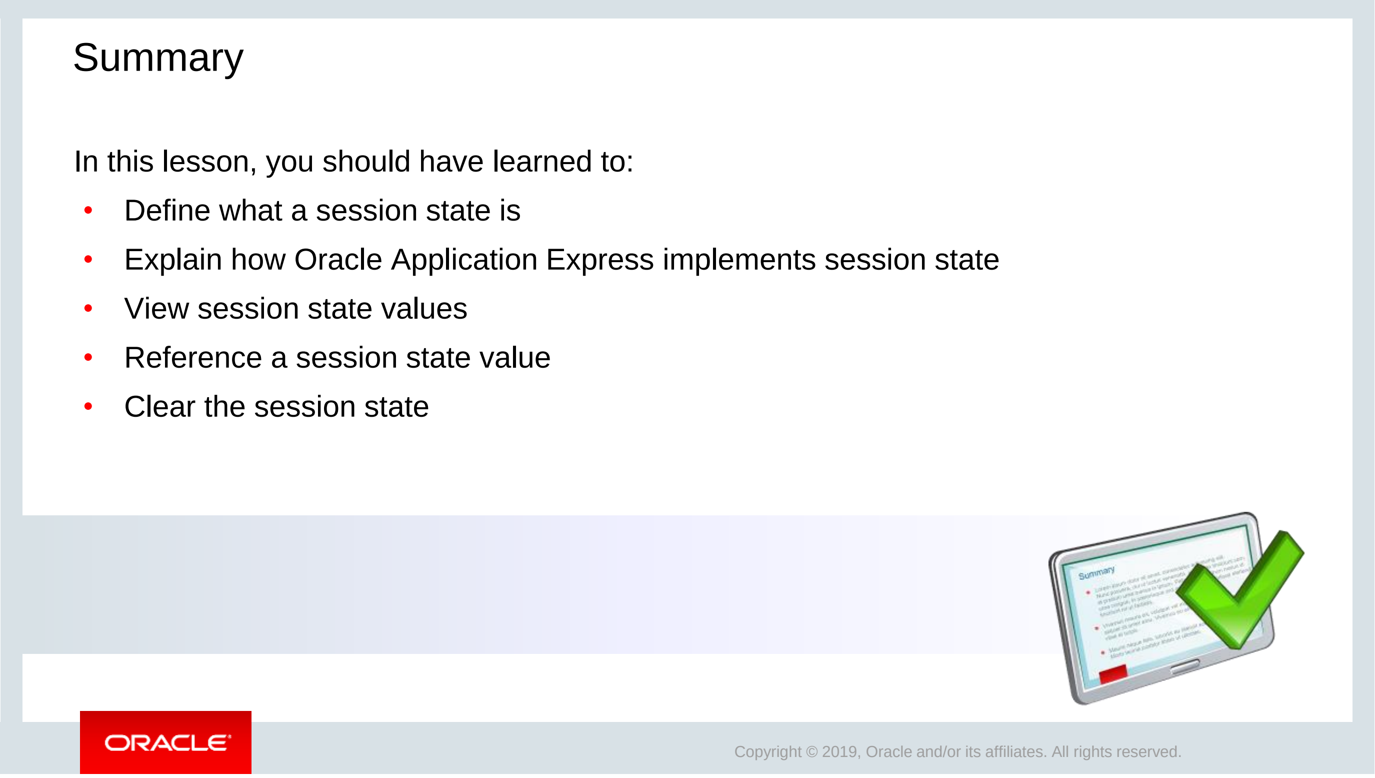
Answer: b



Practice 10-1 Overview: Understanding Session State

This practice covers how session state variables in a page work.

ORACLE



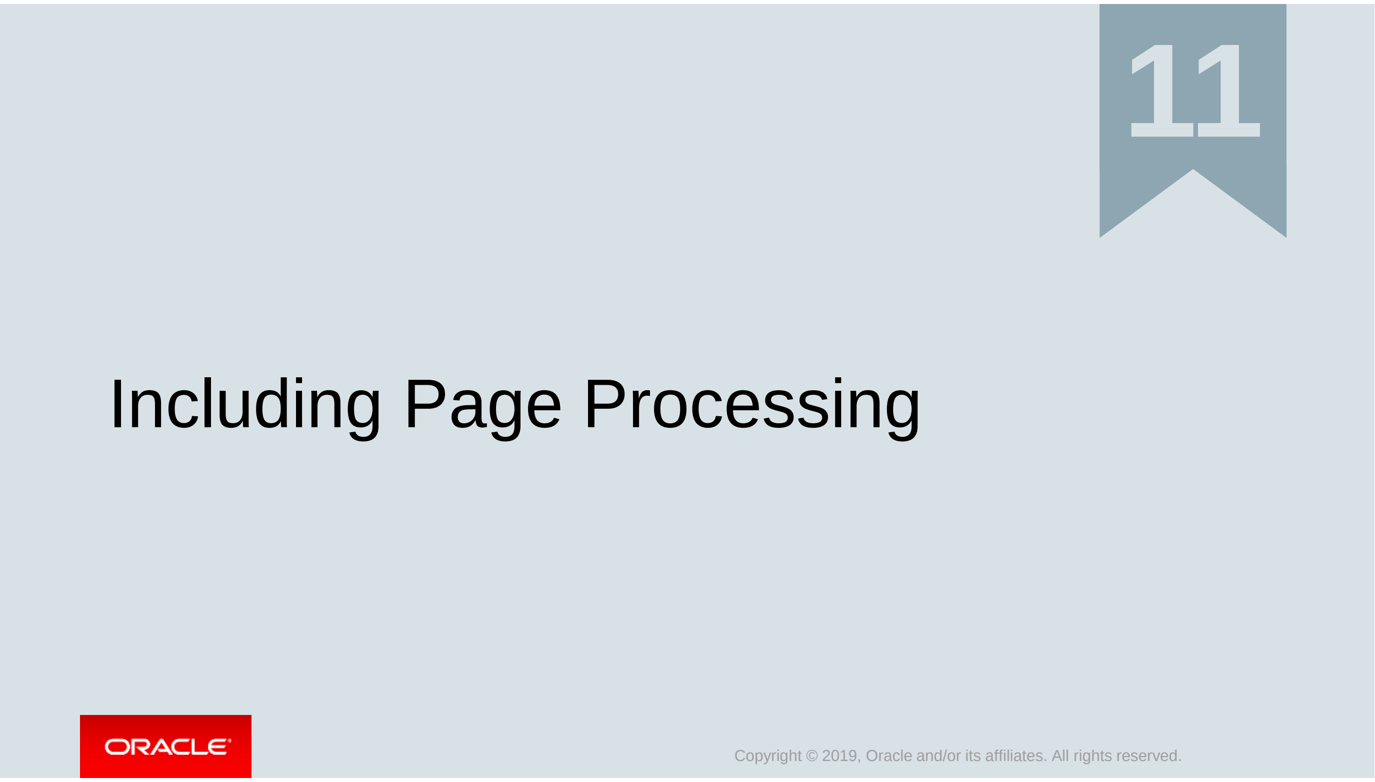
Summary

In this lesson, you should have learned to:

- Define what a session state is
- Explain how Oracle Application Express implements session state
- View session state values
- Reference a session state value
- Clear the session state

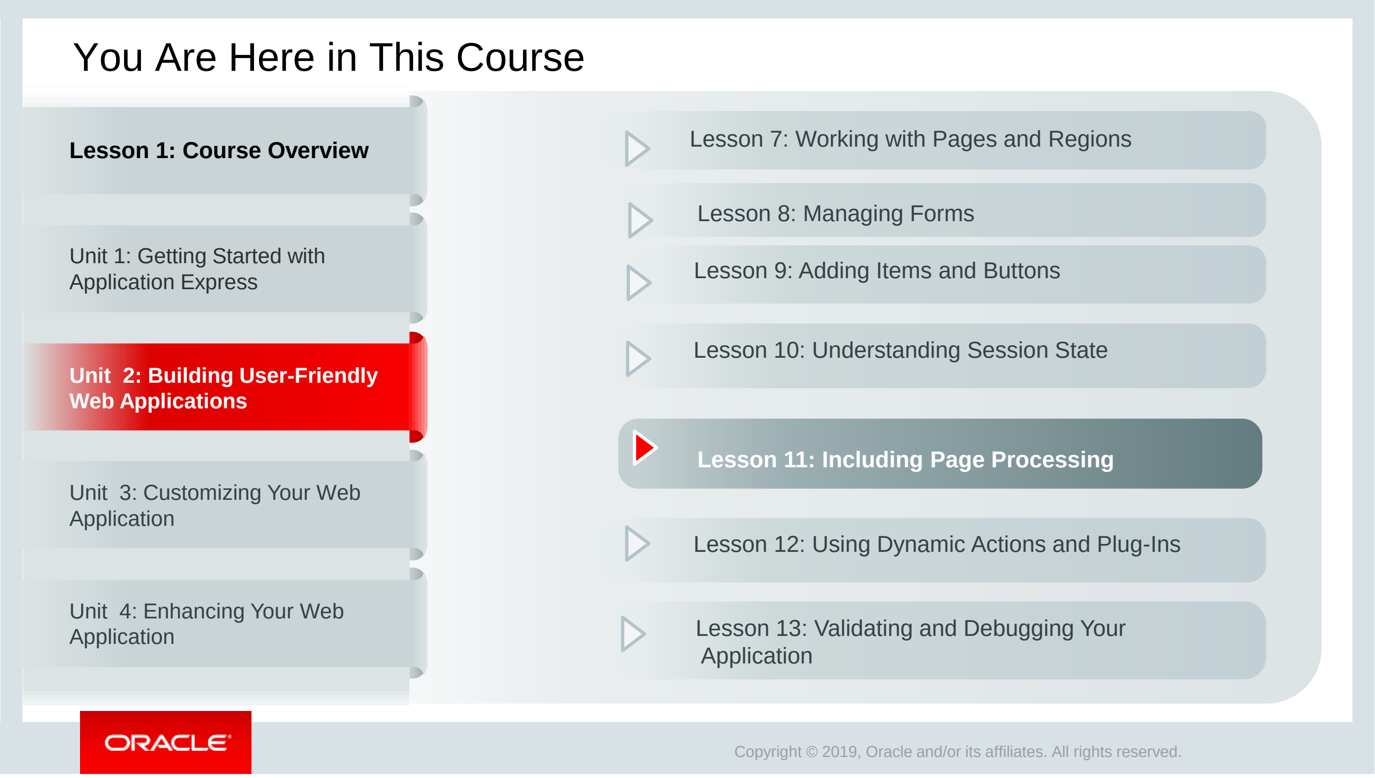
ORACLE

In this lesson, you learned how Oracle Application Express manages the session state of an application. You also learned how to debug an application.



11

Including Page Processing



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

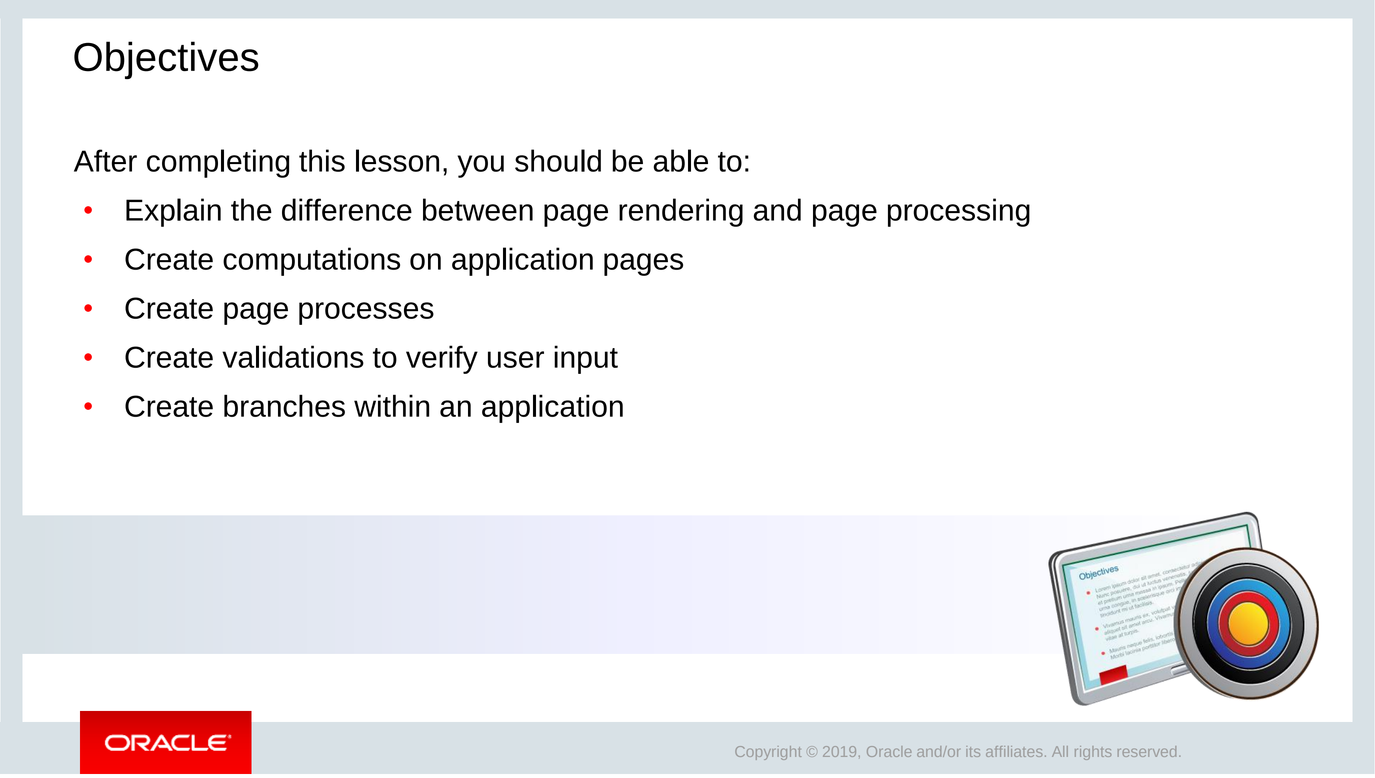
Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- Lesson 7: Working with Pages and Regions
- Lesson 8: Managing Forms
- Lesson 9: Adding Items and Buttons
- Lesson 10: Understanding Session State
- **Lesson 11: Including Page Processing**
- Lesson 12: Using Dynamic Actions and Plug-Ins
- Lesson 13: Validating and Debugging Your Application

This slide is a graphical depiction of the course, particularly highlighting Unit 2 - Lesson 11, which is dealt with in these slides.



Objectives

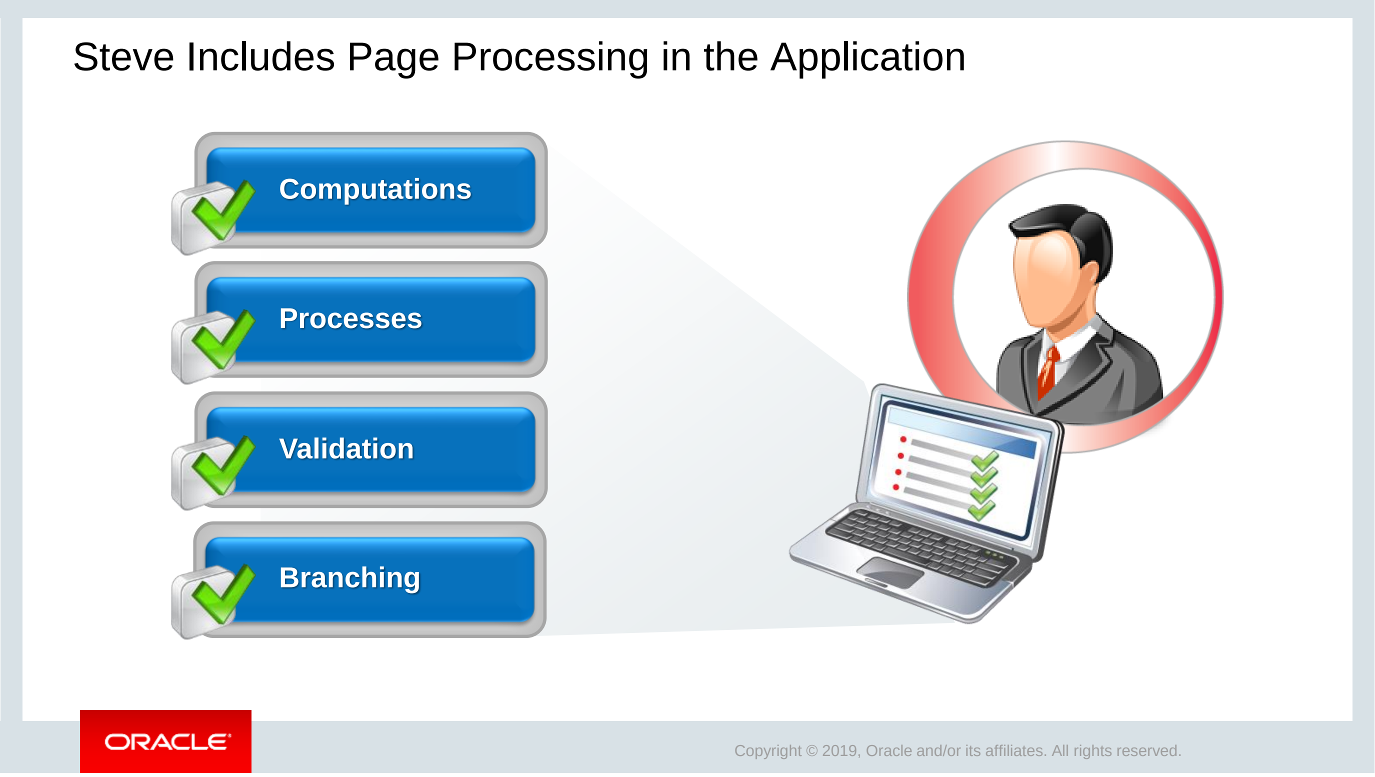
After completing this lesson, you should be able to:

- Explain the difference between page rendering and page processing
- Create computations on application pages
- Create page processes
- Create validations to verify user input
- Create branches within an application

This lesson explains how the Oracle Application Express engine renders and processes a page. You create computations, validations, and processes that are executed when the page is processed. And you create page branches to enable navigation between pages after processing.

- A **page process** performs an action at a specified point during the rendering or submission of the page.
- **Application processes** are blocks of PL/SQL logic that are set to run at specific points using processes from multiple pages of an application.
- **Computations** enable you to assign values to items.
- **Validations** are edit checks on specific items, pages, or columns.
- A **Branch** is an instruction to go to a specific page, procedure, or URL. For example, you can branch from page 1 to page 2 after page 1 is submitted. When you create a branch, you specify a Branch Point and Branch Type.

You will learn in detail about all these features in the following slides.



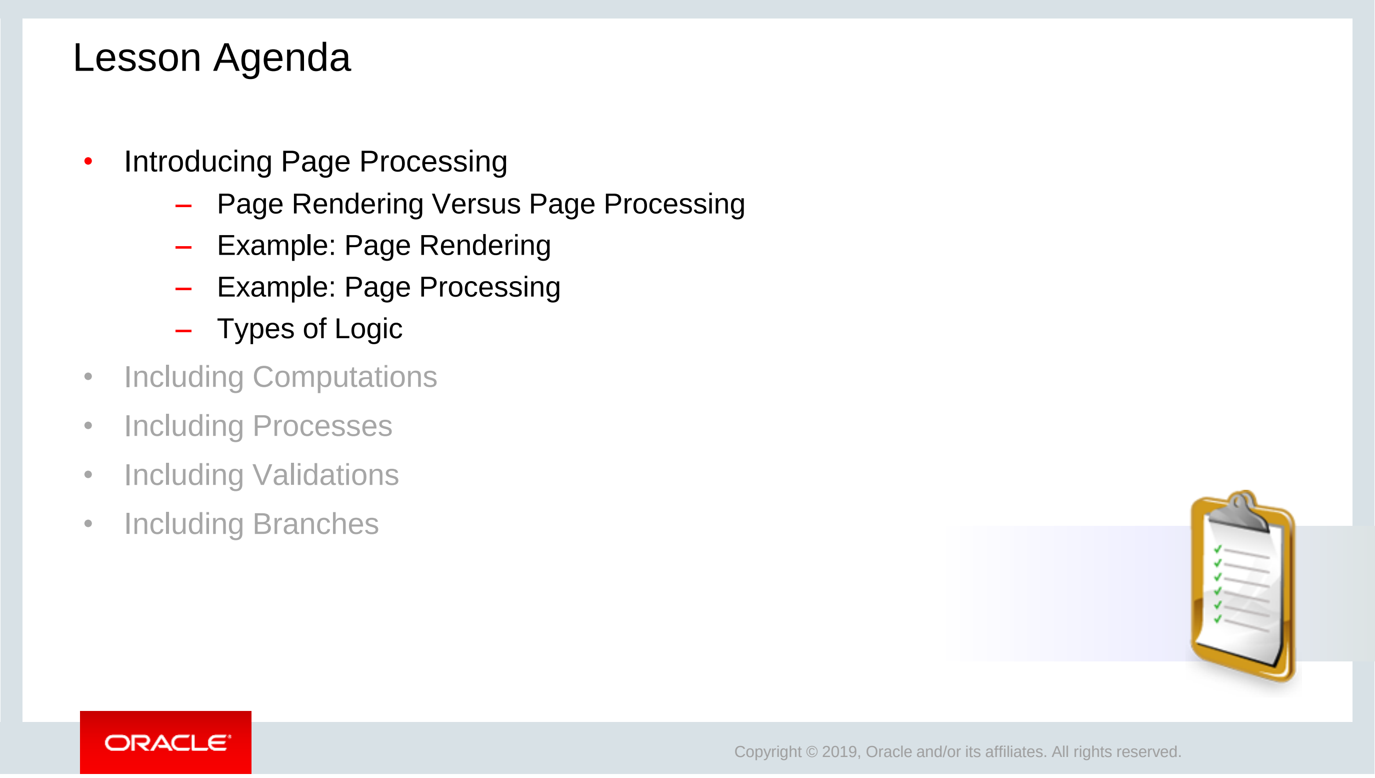
Steve Includes Page Processing in the Application

- Computations
- Processes
- Validation
- Branching

Steve has developed his *Project Tracking System (PTS)* application with reports, forms, and pages, making it functional for the end user. However, the *PTS* application has the following issues:

- It allows the user to enter irrelevant values in a field that expects certain type of values (for example, entering alphabets for phone number).
- It lacks continuity as the pages are not linked. They exist as individual entities.
- It does not have computational ability (that is, it doesn't perform any calculations).

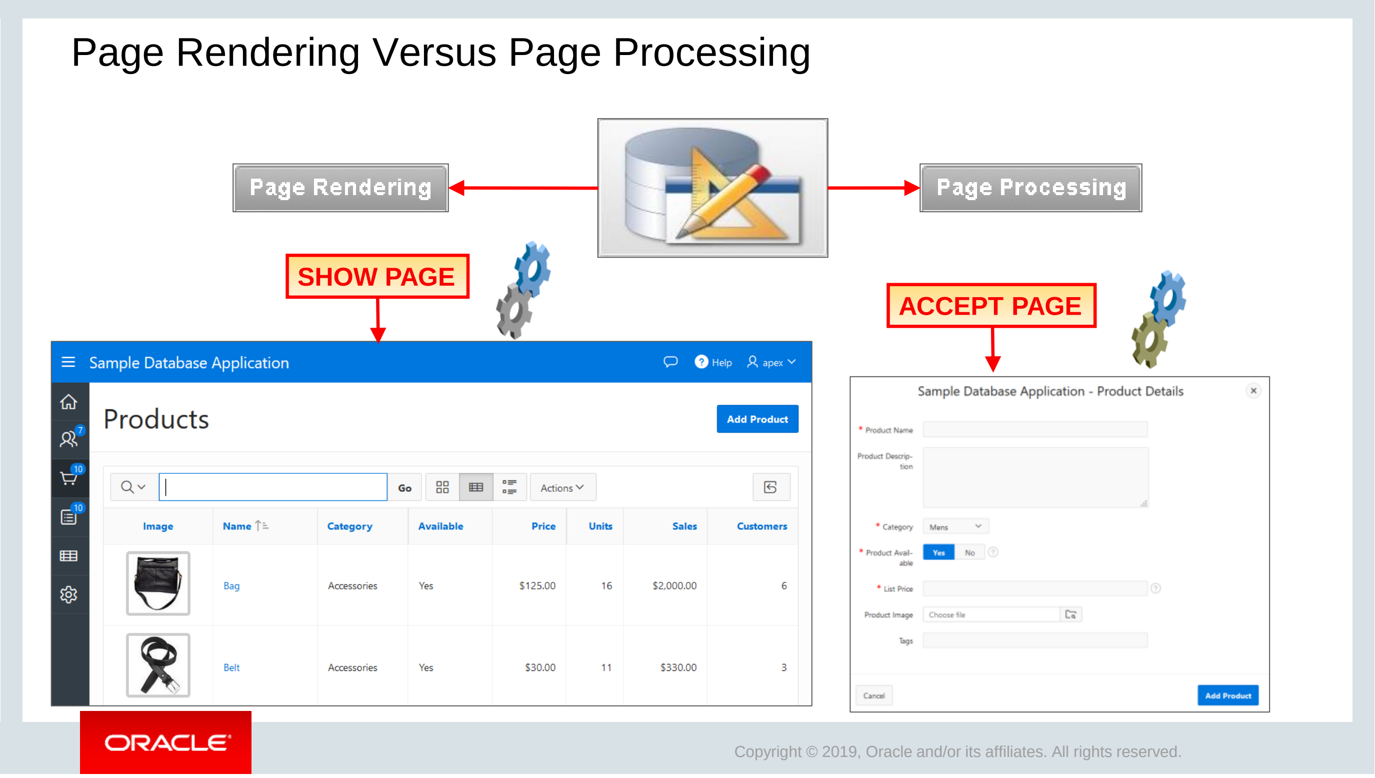
Steve now wants to solve these issues in the application by using the in-built features (like *Computations*, *Processes*, *Validations, and Branching*) of Oracle Application Express.



Lesson Agenda

- Introducing Page Processing
  - Page Rendering Versus Page Processing
  - Example: Page Rendering
  - Example: Page Processing
  - Types of Logic
- Including Computations
- Including Processes
- Including Validations
- Including Branches

ORACLE



Page Rendering Versus Page Processing

Sample Database Application

Help apex

Products

Add Product

Go

Actions

Image	Name	Category	Available	Price	Units	Sales	Customers
	Bag	Accessories	Yes	$125.00	16	$2,000.00	6
	Belt	Accessories	Yes	$30.00	11	$330.00	3

Sample Database Application - Product Details

* Product Name	
Product Description	
* Category	Mens
* Product Available	Yes No
* List Price	
Product Image	Choose file
Tags	

Cancel

Add Product

Do you find any difference in the two images above?

The image on the left shows that when you click the left navigation button (in the *Sample Database Application*) to be displayed on the Products page, the *Products* page is rendered. The right-hand image shows a Forms page. And when you click the **Add Product** button on the Forms page to submit the data, page processing takes place.

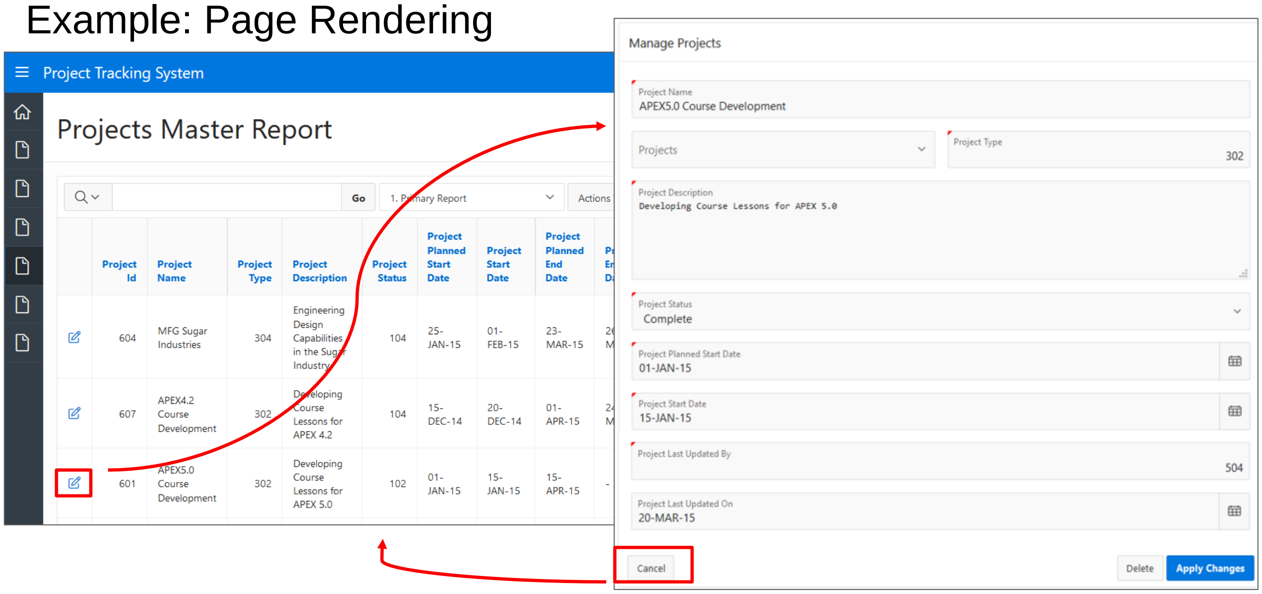
So, how does Oracle Application Express perform *page rendering* and *page processing*? The Application Express engine dynamically renders and processes pages based on data stored in Oracle database tables. To view a rendered version of your application, you request it from the Application Express engine with a URL.

When you run an application, the Application Express engine relies on two processes:

- **Show Page:** Show Page is a page rendering process that assembles all the page attributes (including regions, items, and buttons) into a viewable HTML page. When you request a page using a URL, the engine is running *Show* Page. In the above example, you requested to *SHOW* the Products page.
- **Accept Page:** Accept Page performs forms page processing, including computations, validations, processes, and branching. When you submit a page, the Application Express engine is running Accept Page or performing page processing during which it saves the submitted values in the session cache and then performs any computations, validations, or processes. In the above example, when you enter the product details into the form and click **Add Product**, the details are submitted to the database. Here Oracle Application Express is running *Accept* Page.

In the next few slides, we will see some more examples of *page processing* and *page rendering* to have a clearer understanding.

Example: Page Rendering



Project Tracking System

Projects Master Report

	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Pr En Da
✎	604	MFG Sugar Industries	304	Engineering Design Capabilities in the Sugar Industry	104	25-JAN-15	01-FEB-15	23-MAR-15	26 M
✎	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14	01-APR-15	24 M
✎	601	APEX5.0 Course Development	302	Developing Course Lessons for APEX 5.0	102	01-JAN-15	15-JAN-15	15-APR-15	-

Manage Projects

Project Name: APEX5.0 Course Development

Projects

Project Type: 302

Project Description: `Developing Course Lessons for APEX 5.0`

Project Status: Complete

Project Planned Start Date: 01-JAN-15

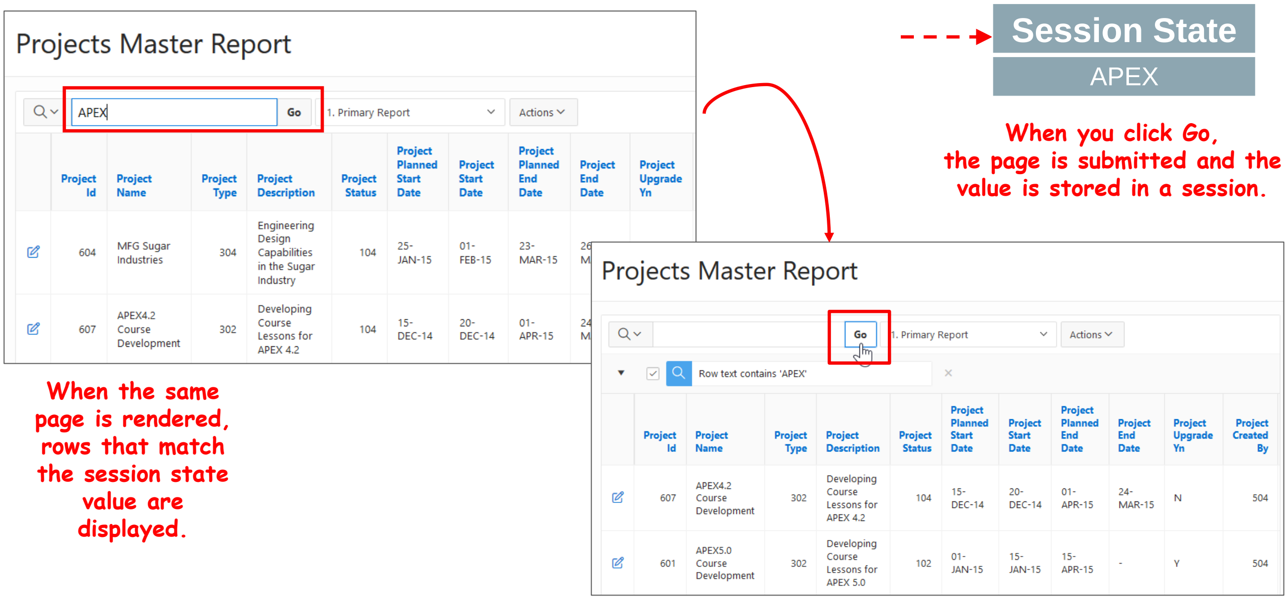
Project Start Date: 15-JAN-15

Project Last Updated By: 504

Project Last Updated On: 20-MAR-15

As you learned in the previous slide, *Page rendering* occurs when the Oracle Application Express engine assembles a page from the database by using a *Show Page* process. See the example on this slide – when you click in the left navigation window of your application page, the *Projects Master Report* page is rendered. Here one page redirects to another page. When you click the Edit icon (pencil symbol) on the left-most column of any record, the page is submitted, and a branch to a form page is invoked. However, when you click the **Cancel** button on the form page, you are redirected to the previous page. Here the page is rendered, but not processed. Since nothing is submitted, so there is no page processing.

Example: Page Processing



Projects Master Report

	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Project End Date	Project Upgrade Yn
	604	MFG Sugar Industries	304	Engineering Design Capabilities in the Sugar Industry	104	25-JAN-15	01-FEB-15	23-MAR-15	26 M	
	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14	01-APR-15	24 M	

Session State

APEX

When you click Go, the page is submitted and the value is stored in a session.

Projects Master Report

☑ Row text contains 'APEX'

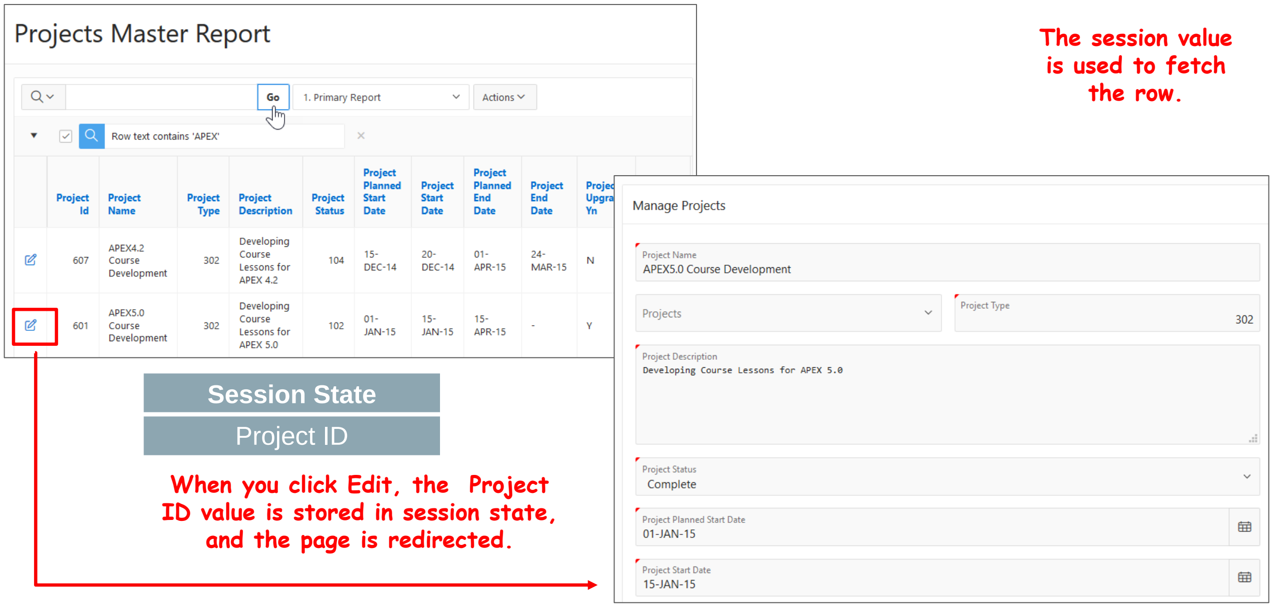
	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Project End Date	Project Upgrade Yn	Project Created By
	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14	01-APR-15	24-MAR-15	N	504
	601	APEX5.0 Course Development	302	Developing Course Lessons for APEX 5.0	102	01-JAN-15	15-JAN-15	15-APR-15	-	Y	504

When the same page is rendered, rows that match the session state value are displayed.

In this slide example, you have a report with a search bar. When you enter a search criterion (here, APEX) and click the **Go** button, the page is submitted. A process stores the search value in a session state, and a branch to the same page is invoked. When the page is displayed again, a process runs to display only those rows that match the value stored in the session state. Therefore, here page processing takes place.

So you see that *Page processing* occurs when the Oracle Application Express engine executes a process by using the data submitted from a page. For page processing, the Oracle Application Express engine runs an *Accept Page* process.

Example: Page Processing



Projects Master Report

	Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Project End Date	Projec Upgra Yn
	607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14	01-APR-15	24-MAR-15	N
	601	APEX5.0 Course Development	302	Developing Course Lessons for APEX 5.0	102	01-JAN-15	15-JAN-15	15-APR-15	-	Y

Manage Projects

Project Name: APEX5.0 Course Development

Projects

Project Type: 302

Project Description: Developing Course Lessons for APEX 5.0

Project Status: Complete

Project Planned Start Date: 01-JAN-15

Project Start Date: 15-JAN-15

The session value is used to fetch the row.

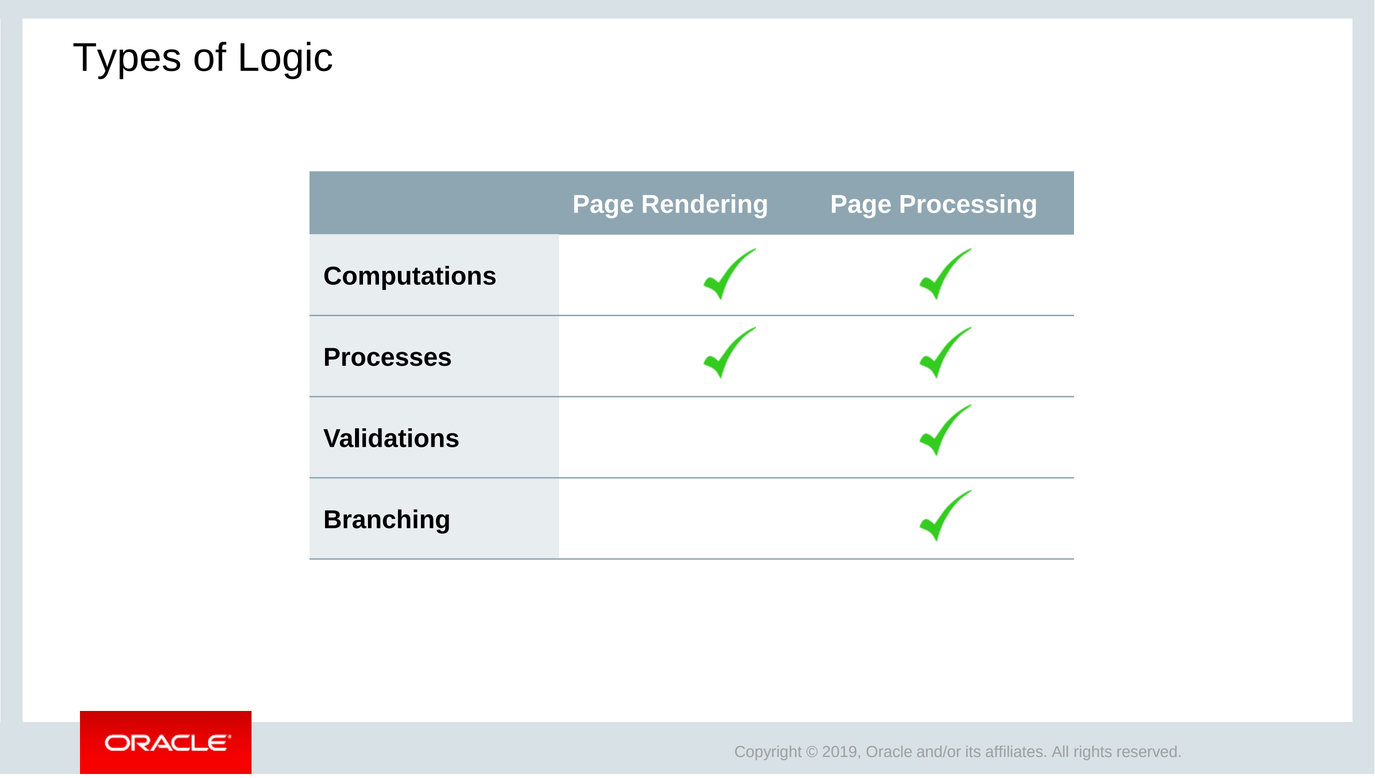
Session State

Project ID

When you click Edit, the Project ID value is stored in session state, and the page is redirected.

Let’s see another scenario to understand *page processing*. In this example, you have an editable reports page. When you click the Edit icon for a row in the report, the ID value for the row is stored in session state, and you are redirected to a forms page. When the form page is displayed, a process runs to fetch the row details by using the ID value stored in the session state.

Typically, a page is submitted when a user clicks a button. You use the Page Processing section of page definition to specify application logic such as computations, validations, processes, and branches.

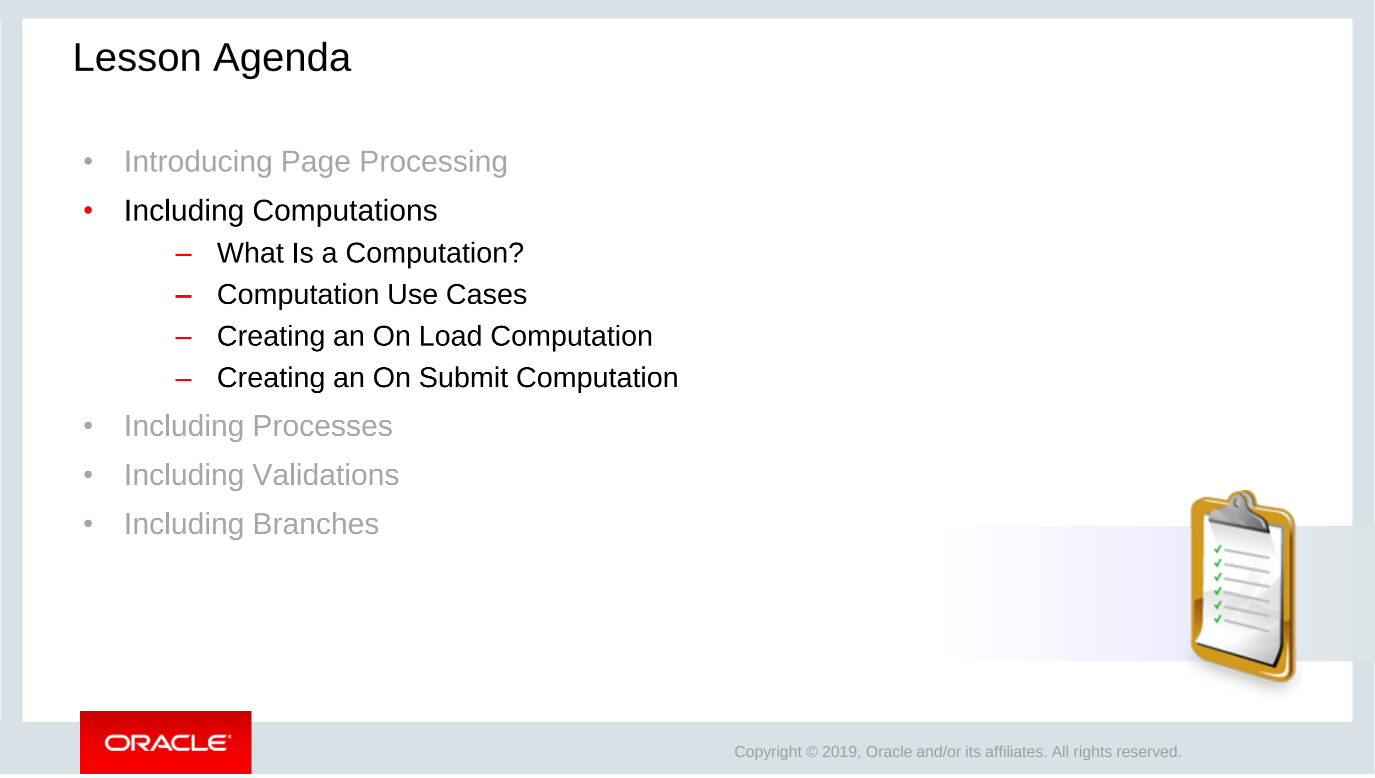


Types of Logic

	Page Rendering	Page Processing
Computations	✓	✓
Processes	✓	✓
Validations		✓
Branching		✓

You learned in the previous slide that there are four types of logic that you can perform on a page: computations, processes, validations, and branching. When you are creating the logic, you can specify at which point you want to perform logic. If you have more than one process or computation defined at the same point, you can specify a sequence.

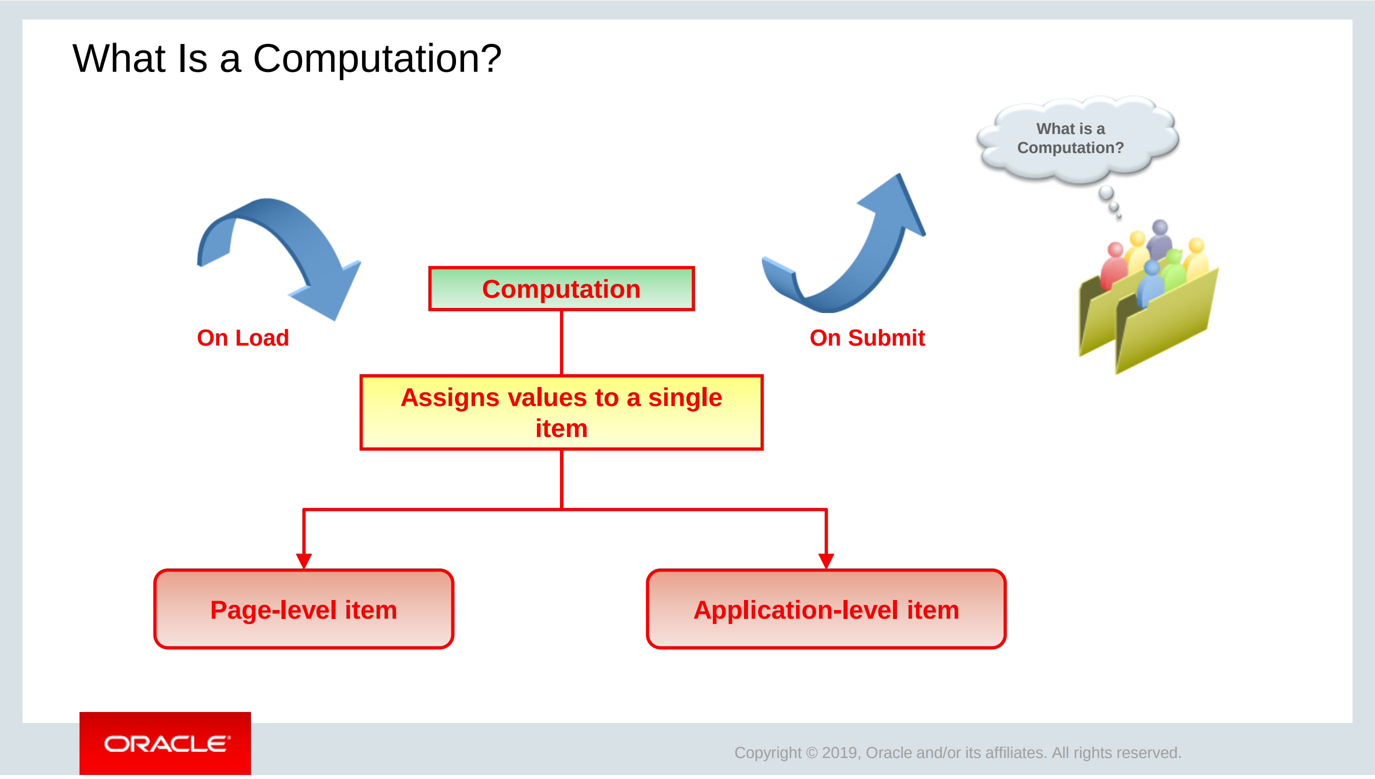
Page rendering computations and processes are performed when the HTML page is assembled and displayed, whereas page processing computations and processes are performed when the page is submitted to the Oracle Application Express engine. You will learn about each of them in detail in the next few slides.



Lesson Agenda

- Introducing Page Processing
- Including Computations
  - What Is a Computation?
  - Computation Use Cases
  - Creating an On Load Computation
  - Creating an On Submit Computation
- Including Processes
- Including Validations
- Including Branches

ORACLE



What Is a Computation?

What is a Computation?

On Load

Computation

On Submit

Assigns values to a single item

Page-level item

Application-level item

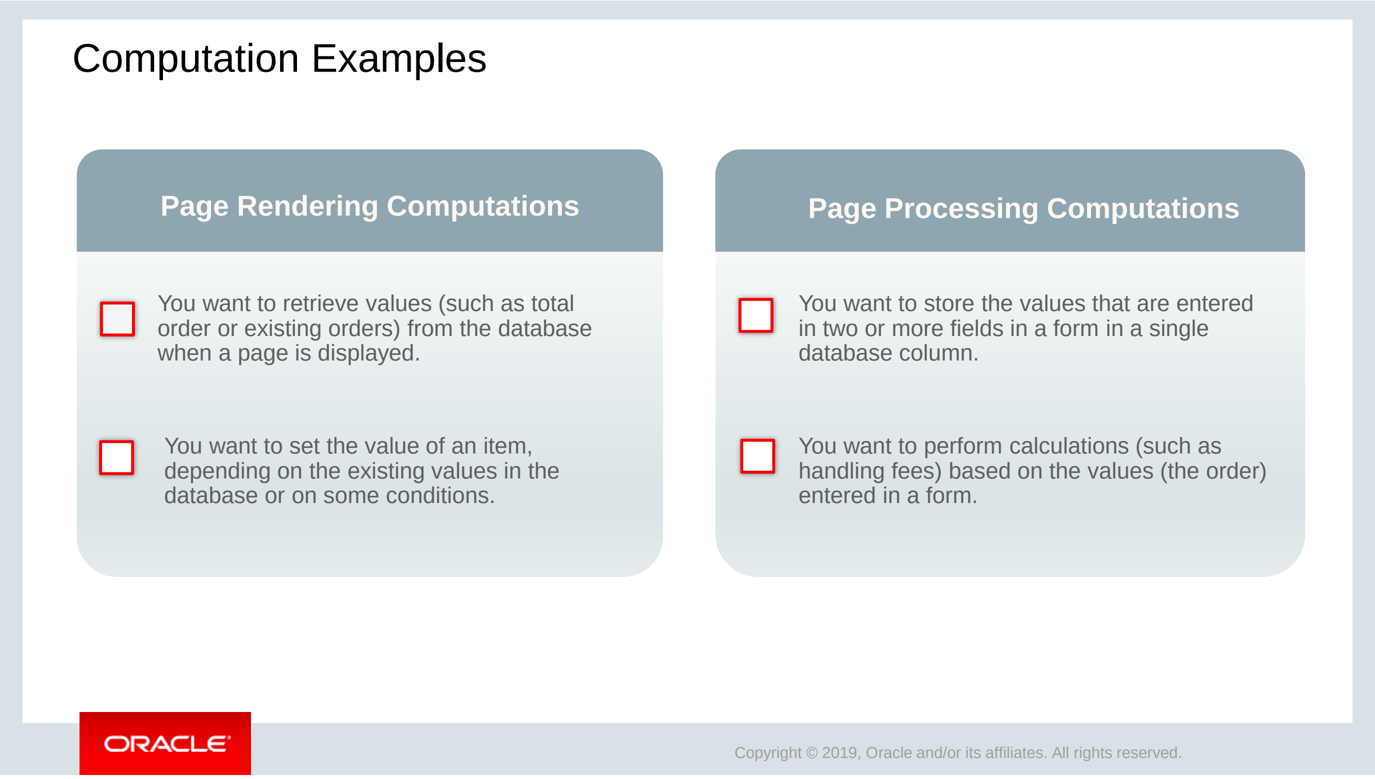
ORACLE

Now Steve wants to use a logic that changes the employee’s *Email ID* to uppercase after the page is submitted and before the value is inserted/updated into the table. He wants to specify that whenever the end user is entering the email address, it should be stored in the database (`EMPLOYEES` table) in uppercase. Let’s see how he uses Computation to do so.

You learned about page items and application items in the lesson on *Items and Buttons*. Let’s try recollecting what are page items and application items.

Page-level items are placed on a page and have associated user interface properties, but *application-level items* are not associated with a page and, therefore, have no user interface properties. You can use an application item as a global variable.

A computation is a logic that assigns values to a single item. *Application-level computations* assign a value to an application item when any page in an application is rendered or processed. A typical use of application computation is to store the number of the last page visited. In contrast, *page-level computations* assign a value to an identified item when a page is displayed or submitted (rendered or processed).



Computation Examples

Page Rendering Computations

□ You want to retrieve values (such as total order or existing orders) from the database when a page is displayed.

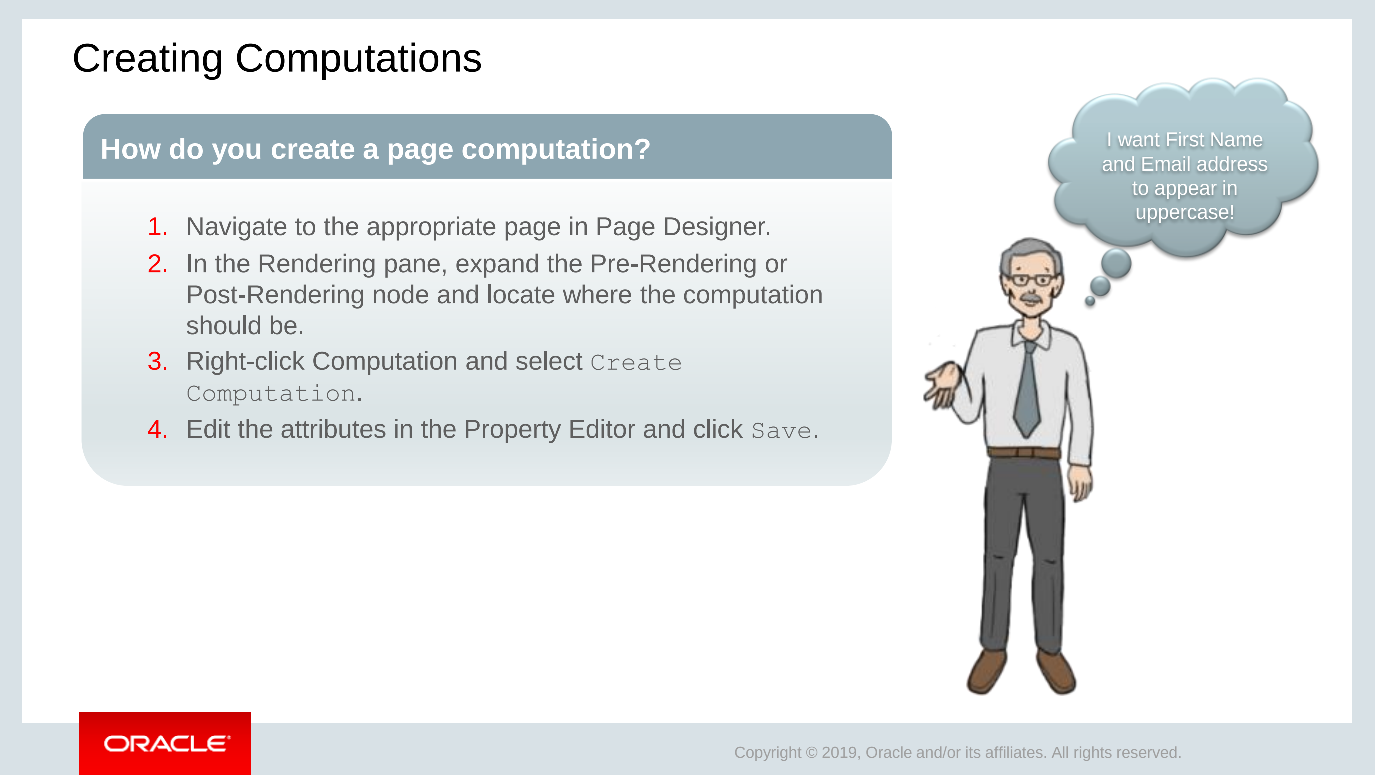
□ You want to set the value of an item, depending on the existing values in the database or on some conditions.

Page Processing Computations

□ You want to store the values that are entered in two or more fields in a form in a single database column.

□ You want to perform calculations (such as handling fees) based on the values (the order) entered in a form.

You learned that you can create computations that are executed when a page is rendered or when a page is processed. This slide lists some scenarios when you can create page rendering or page-processing computations. Let’s learn more in the next few slides.



Creating Computations

How do you create a page computation?

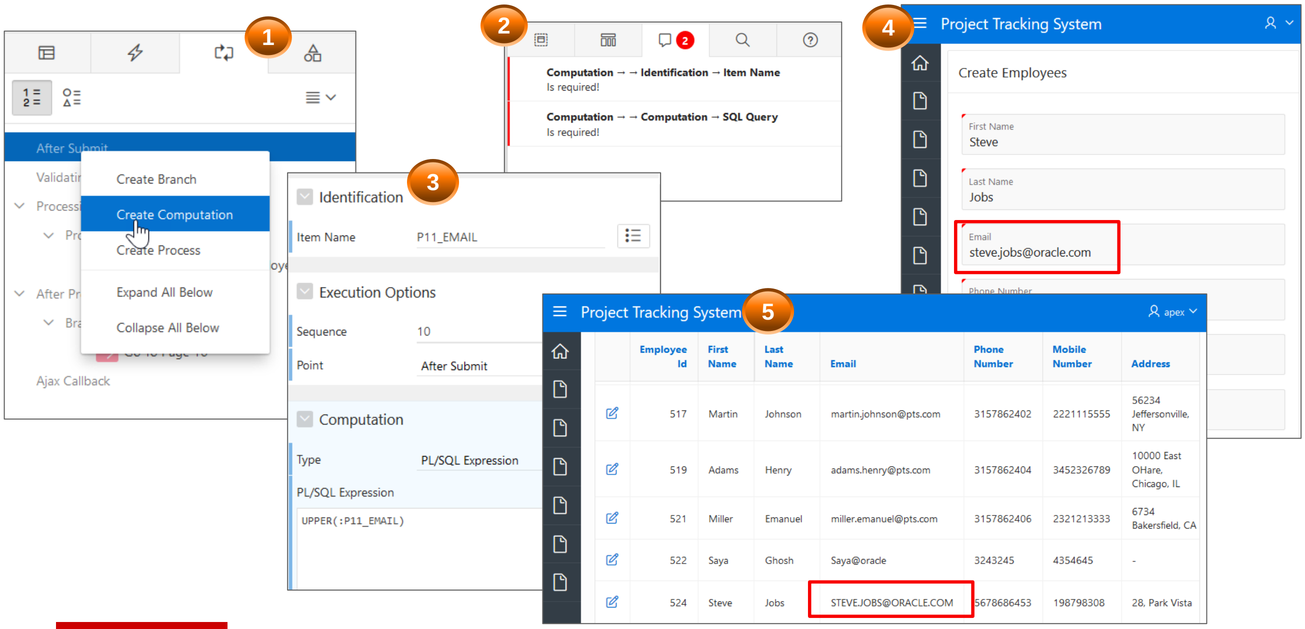
1. Navigate to the appropriate page in Page Designer.
2. In the Rendering pane, expand the Pre-Rendering or Post-Rendering node and locate where the computation should be.
3. Right-click Computation and select `Create Computation`.
4. Edit the attributes in the Property Editor and click `Save`.

I want First Name and Email address to appear in uppercase!

ORACLE

This slide provides an overview of the steps to create a page computation. In the next slide, you will see how Steve converts email address into uppercases using computation.

Example: Creating Page-Processing Computations



Computation → → Identification → Item Name
Is required!

Computation → → Computation → SQL Query
Is required!

Identification

Item Name: P11_EMAIL

Execution Options

Sequence: 10

Point: After Submit

Computation

Type: PL/SQL Expression

PL/SQL Expression

```
UPPER(:P11_EMAIL)
```

Create Employees

First Name: Steve

Last Name: Jobs

Email: steve.jobs@oracle.com

Phone Number

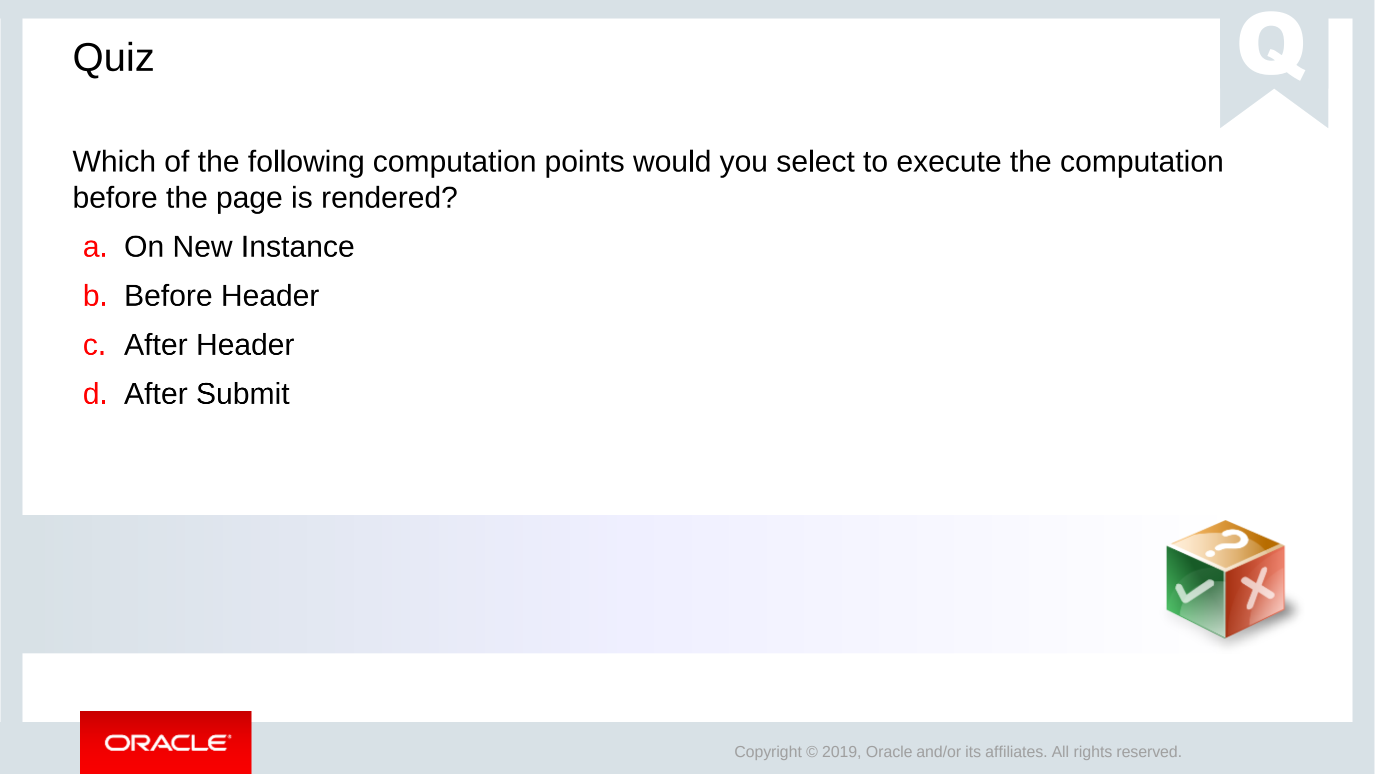
	Employee Id	First Name	Last Name	Email	Phone Number	Mobile Number	Address
✎	517	Martin	Johnson	martin.johnson@pts.com	3157862402	2221115555	56234 Jeffersonville, NY
✎	519	Adams	Henry	adams.henry@pts.com	3157862404	3452326789	10000 East OHare, Chicago, IL
✎	521	Miller	Emanuel	miller.emanuel@pts.com	3157862406	2321213333	6734 Bakersfield, CA
✎	522	Saya	Ghosh	Saya@oracle	3243245	4354645	-
✎	524	Steve	Jobs	STEVE.JOBS@ORACLE.COM	5678686453	198798308	28, Park Vista

Now let's see how Steve creates a computation that changes the *Email ID* to uppercase after the page is submitted and before it gets inserted/updated into the database table:

1. Navigate to the appropriate page in Page Designer. Here, Steve selects **Create Employees** form page.
2. Click the Processing button of the left pane.
3. Under Processing, right-click **After Submit** and select **Create Computation** to create a new page processing computation (screenshot 1). The computation is created, and the attributes of the computation are displayed in the Property Editor (screenshot 3).

3. Click the Messages tab (screenshot 2) to resolve the errors. To do so, in the Property Editor, edit the following attributes and click **Save**.
   - **Identification: Item Name**: Select (from Popup LOV) `P11_EMAIL`
   - **Computation: Type**: Select *PL/SQL Expression*
   - **PL/SQL Expression** (code text area): Enter `UPPER(:P11_EMAIL)`

5. Click **Save** to create the computation. The computation is created and is listed under the **Computations** node.
6. Enter details for a new employee with email ID in lower case (in this example, we have added '*steve.jobs@oracle.com*') (screenshot 4) and click **Create**. You can see the email for the new row shows up in upper case on the report immediately (screenshot 5).



Quiz

Which of the following computation points would you select to execute the computation before the page is rendered?

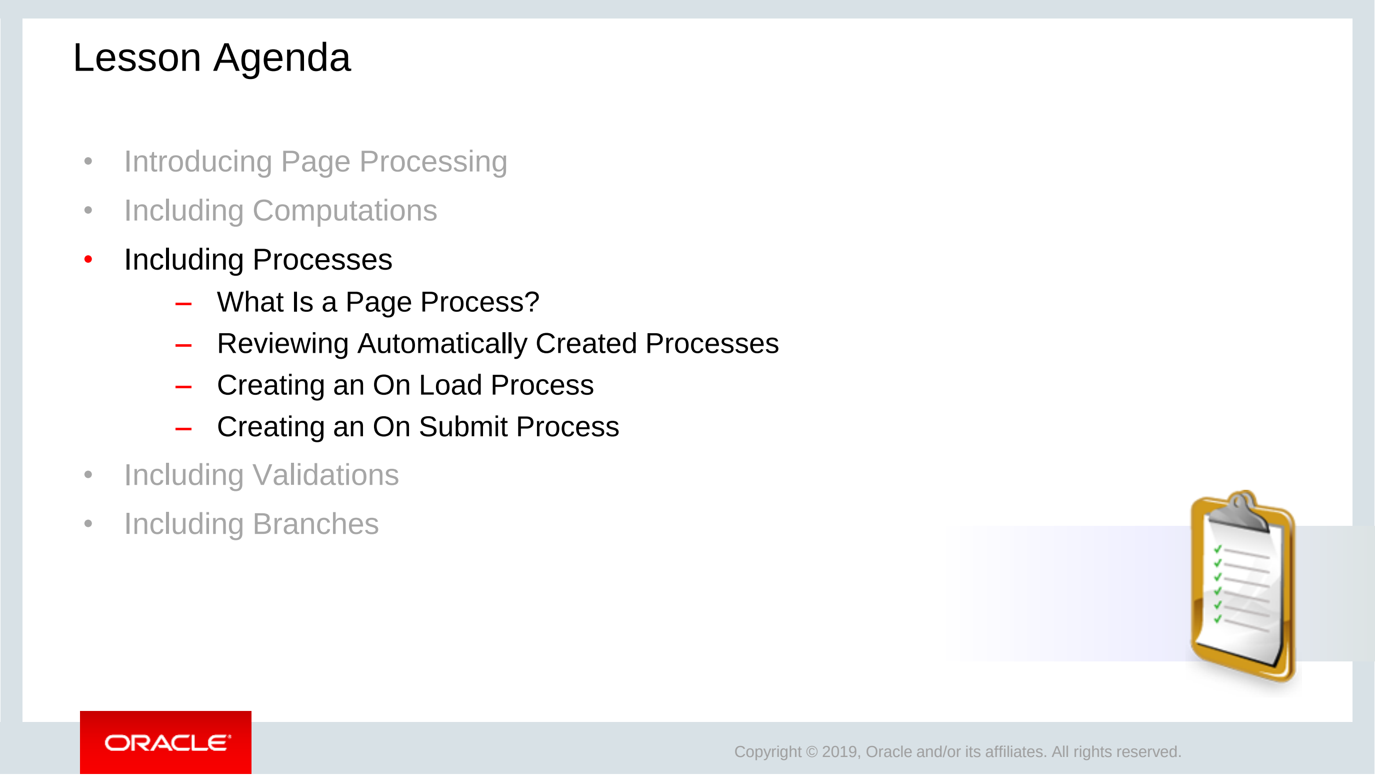
a. On New Instance

b. Before Header

c. After Header

d. After Submit

Answer: b and c

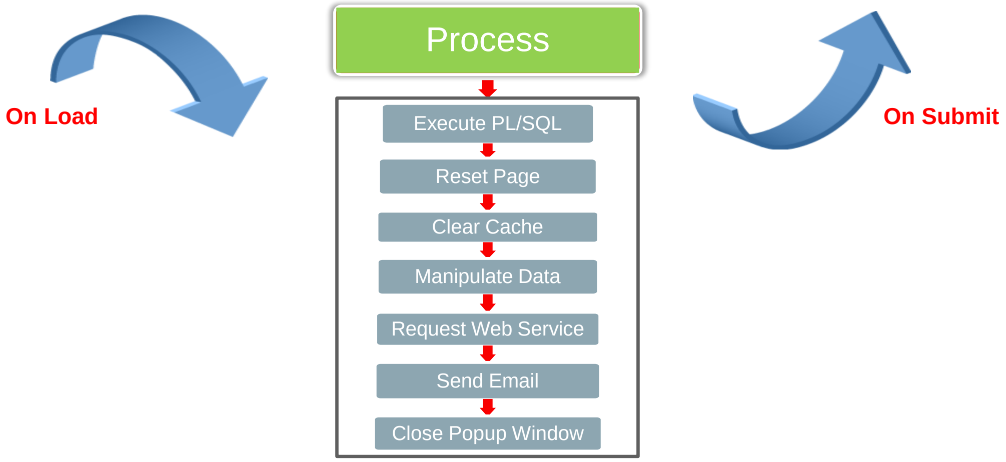


Lesson Agenda

- Introducing Page Processing
- Including Computations
- Including Processes
  - What Is a Page Process?
  - Reviewing Automatically Created Processes
  - Creating an On Load Process
  - Creating an On Submit Process
- Including Validations
- Including Branches

What Is a Page Process?

A page process is used to perform a specific action when a page is rendered or submitted.



On Load

Process

Execute PL/SQL

Reset Page

Clear Cache

Manipulate Data

Request Web Service

Send Email

Close Popup Window

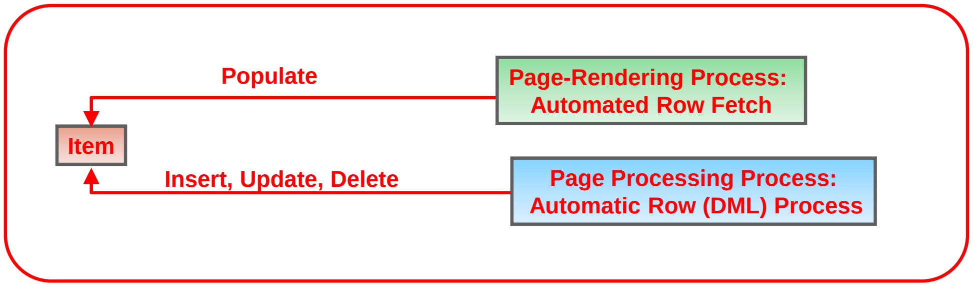
On Submit

Like computation, *page process* is also a unit of logic that runs when a specific event occurs, such as loading or submitting a page. You can create a page process to execute some code (such as SQL or PL/SQL) or to make a call to the Application Express engine. When you use wizards, such as *Create Report* or *Create Form*, some processes are automatically created. For example, a process to insert, update, or delete a row from the database is created when the user clicks the appropriate button. The next few slides discuss some automatically created processes.

Automatic Processing Processes

Oracle Application Express provides automatic data manipulation language (DML) processing.

- You are not required to provide any SQL code; just reference a database column.
- You can use it to update or delete rows of a table.


Populate
Page-Rendering Process: Automated Row Fetch
Item
Insert, Update, Delete
Page Processing Process: Automatic Row (DML) Process

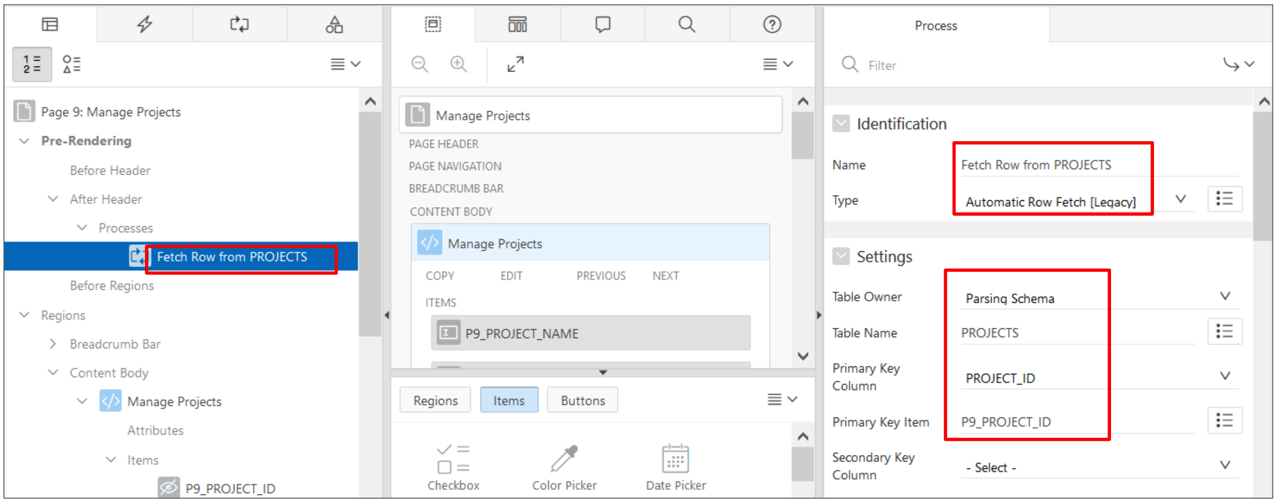
ORACLE

When you create a form by using the Create Form Wizard, the wizard creates two processes:

- The Automated Row Fetch process that is executed when a page is rendered. This process populates the items by fetching data from the database.
- The Automatic Row (DML) process that is executed when a page is submitted. This process updates the database by using the `INSERT`, `DELETE`, or `UPDATE` command.

These processes are automatic in that you must specify only the database column names and not any SQL code. They also perform lost update detection. Lost update detection ensures that data integrity in applications is maintained where data can be accessed concurrently.

Reviewing an Automated Row Fetch Process



Identification

Name	Fetch Row from PROJECTS
Type	Automatic Row Fetch [Legacy]

Settings

Table Owner	Parsing Schema
Table Name	PROJECTS
Primary Key Column	PROJECT_ID
Primary Key Item	P9_PROJECT_ID
Secondary Key Column	- Select -

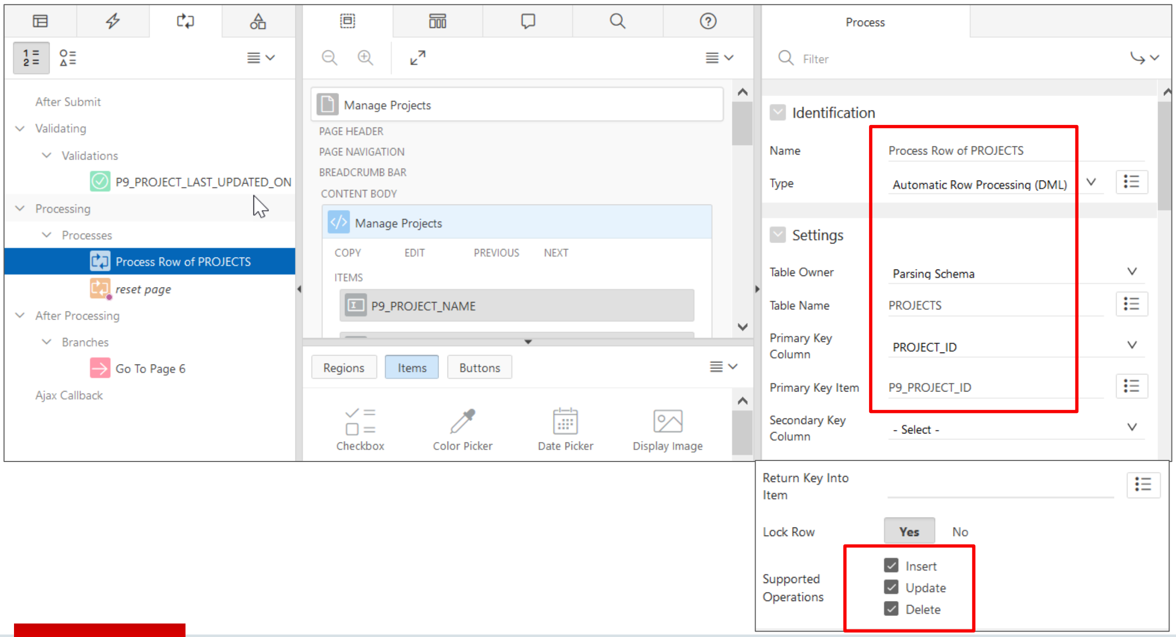
The *Automated Row Fetch* process is executed when a page is rendered. This process populates the items by fetching data from the database. In the above slide, you see that when the *Manage Projects* form page is rendered, it fetches data from the `PROJECTS` table.

An *Automated Row Fetch* process populates the fields in a form by retrieving data from a database table, by using a primary key column value.

To view an Automated Row Fetch process, navigate to the page definition of the page that contains a form created by the *Create Form* Wizard. Perform the following steps:

1. In the Pre-Rendering section, click **Fetch Row from <table_name>** under the *After Header* node.
2. You can view the process details in the Property Editor window. The **Identification** tab shows the *Name* and *Type,* and under the **Settings** tab, the *Table Name*, *Primary Key Column, Secondary Key Column,* and *Primary Key Item* are listed.

Reviewing an Automatic Row (DML) Processing Process



After Submit
- Validating
  - Validations
    - P9_PROJECT_LAST_UPDATED_ON
- Processing
  - Processes
    - Process Row of PROJECTS
    - *reset page*
- After Processing
  - Branches
    - Go To Page 6

Ajax Callback

Manage Projects
PAGE HEADER
PAGE NAVIGATION
BREADCRUMB BAR
CONTENT BODY
Manage Projects
COPY EDIT PREVIOUS NEXT
ITEMS
P9_PROJECT_NAME

Process

Identification

Name	Process Row of PROJECTS
Type	Automatic Row Processing (DML)

Settings

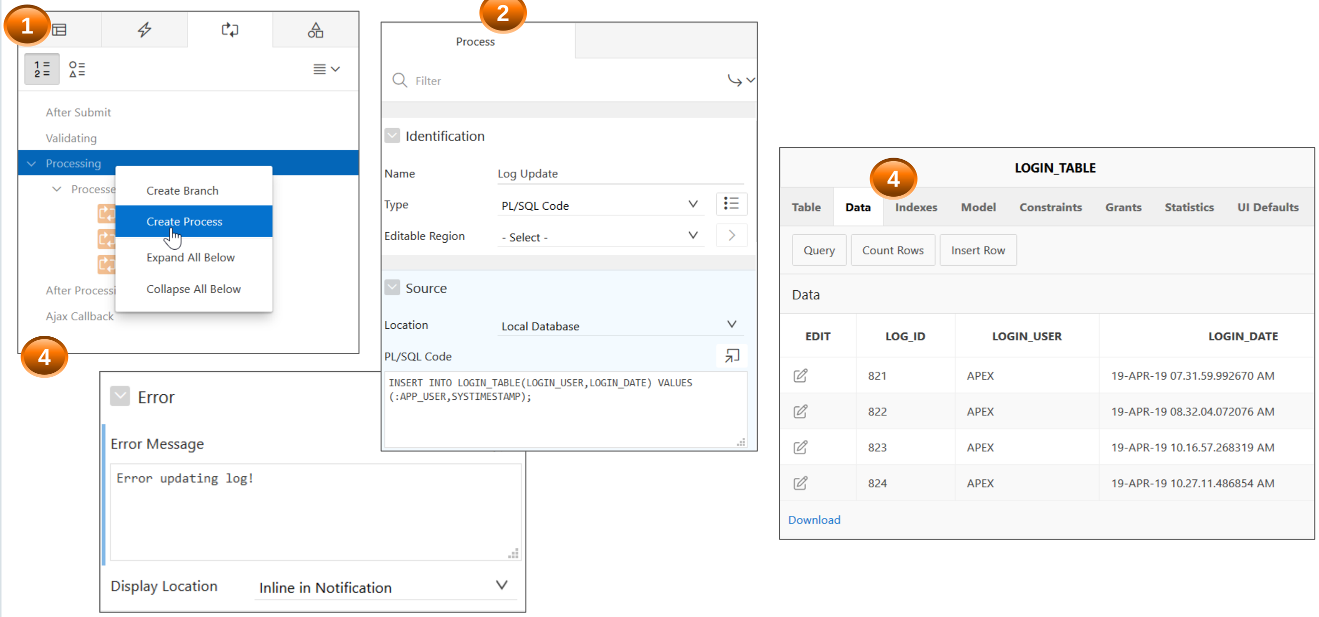
Table Owner	Parsing Schema
Table Name	PROJECTS
Primary Key Column	PROJECT_ID
Primary Key Item	P9_PROJECT_ID
Secondary Key Column	- Select -
Return Key Into Item	
Lock Row	Yes No
Supported Operations	☑ Insert ☑ Update ☑ Delete

The *Automatic Row (DML)* process is executed when a page (in this example, *Manage Projects* form page) is submitted. This process updates the database by using the `INSERT`, `DELETE`, or `UPDATE` commands.

To view an Automated Row (DML) process, navigate to the Processing tab. Perform the following steps:

1. In the left panel of the Page Designer, click **Process Row of <table_name>** under the *Processing* node.
2. You can view the process details in the Property Editor. The *Table Name*, *Primary Key Item*, *Primary Key Column,* and the operations that are allowed on the table are listed.
3. You can view the Supported Operations: **Insert, Update, Delete.**

Creating an On Submit Process



Identification

Name: Log Update

Type: PL/SQL Code

Editable Region: - Select -

Source

Location: Local Database

PL/SQL Code

```
INSERT INTO LOGIN_TABLE(LOGIN_USER,LOGIN_DATE) VALUES
(:APP_USER,SYSTIMESTAMP);
```

Error

Error Message

```
Error updating log!
```

Display Location: Inline in Notification

LOGIN_TABLE

Data

EDIT	LOG_ID	LOGIN_USER	LOGIN_DATE
✎	821	APEX	19-APR-19 07.31.59.992670 AM
✎	822	APEX	19-APR-19 08.32.04.072076 AM
✎	823	APEX	19-APR-19 10.16.57.268319 AM
✎	824	APEX	19-APR-19 10.27.11.486854 AM

Now Steve wants to create an *on Submit* process to log the user details into the database after the *Sign in* Page is submitted. For example, whenever a user logs in to his *PTS* application, the user's name and the login date and time get stored in a database table. Let's see how Steve creates an *on Submit* process that will be invoked automatically after the page is submitted:

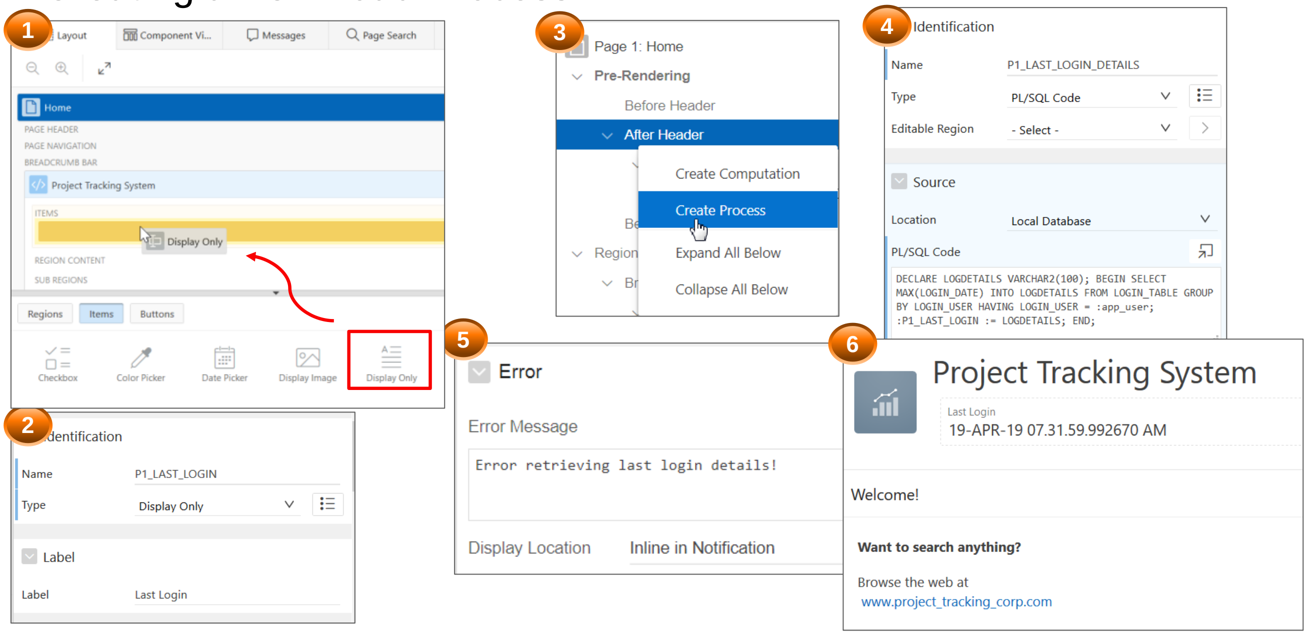
1. Open the **Sign in** (login 9999) page in Page Designer view.
2. Right-click the **Processes** node under **Processing** and select *Create Process*.
3. In the Property Editor pane, enter the following details and click **Save**.
   - **Identification**: **Name**: Enter `Log Update`
   - **Identification**: **Type**: Select **PL/SQL Code**
   - **Source**: **PLSQL Code** (code text area): Enter the PL/SQL Code in the text area. In this example, Steve enters an `INSERT` command to enter the application username (`:APP_USER`) and the date/time information (sysdate) into a `LOG` table:

     ```
     INSERT INTO LOGIN_TABLE(LOGIN_USER,LOGIN_DATE) VALUES
     (:APP_USER,SYSTIMESTAMP);
     ```
   - **Success Message: Success Message**: Enter `Log updated!`
   - **Error**: **Error Message**: Enter `Error updating Log!`

   The process is created.
4. Log out and log in to the application. Check if an entry is made in `LOGIN_TABLE` by clicking **SQL Workshop** and then selecting **Object Browser**.
5. Select **LOGIN_TABLE** *(Data Tab)*, and you see a new record is added with following details:

LOG_ID	***LOGIN_USER***	***LOGIN_DATE***
<some value>	APEX_ADMIN	*<time stamp of login>*

Creating an On Load Process



Name P1_LAST_LOGIN
Type Display Only
Label Last Login

Name P1_LAST_LOGIN_DETAILS
Type PL/SQL Code
Editable Region - Select -
Location Local Database
PL/SQL Code

```
DECLARE LOGDETAILS VARCHAR2(100); BEGIN SELECT
MAX(LOGIN_DATE) INTO LOGDETAILS FROM LOGIN_TABLE GROUP
BY LOGIN_USER HAVING LOGIN_USER = :app_user;
:P1_LAST_LOGIN := LOGDETAILS; END;
```

Error Message

```
Error retrieving last login details!
```

Display Location Inline in Notification

Last Login
19-APR-19 07.31.59.992670 AM

Another requirement in the *PTS* application is to display information about the last login such as date and time. For this, Steve creates an on load process to retrieve the date and time when a user last logged in to the *PTS* application:

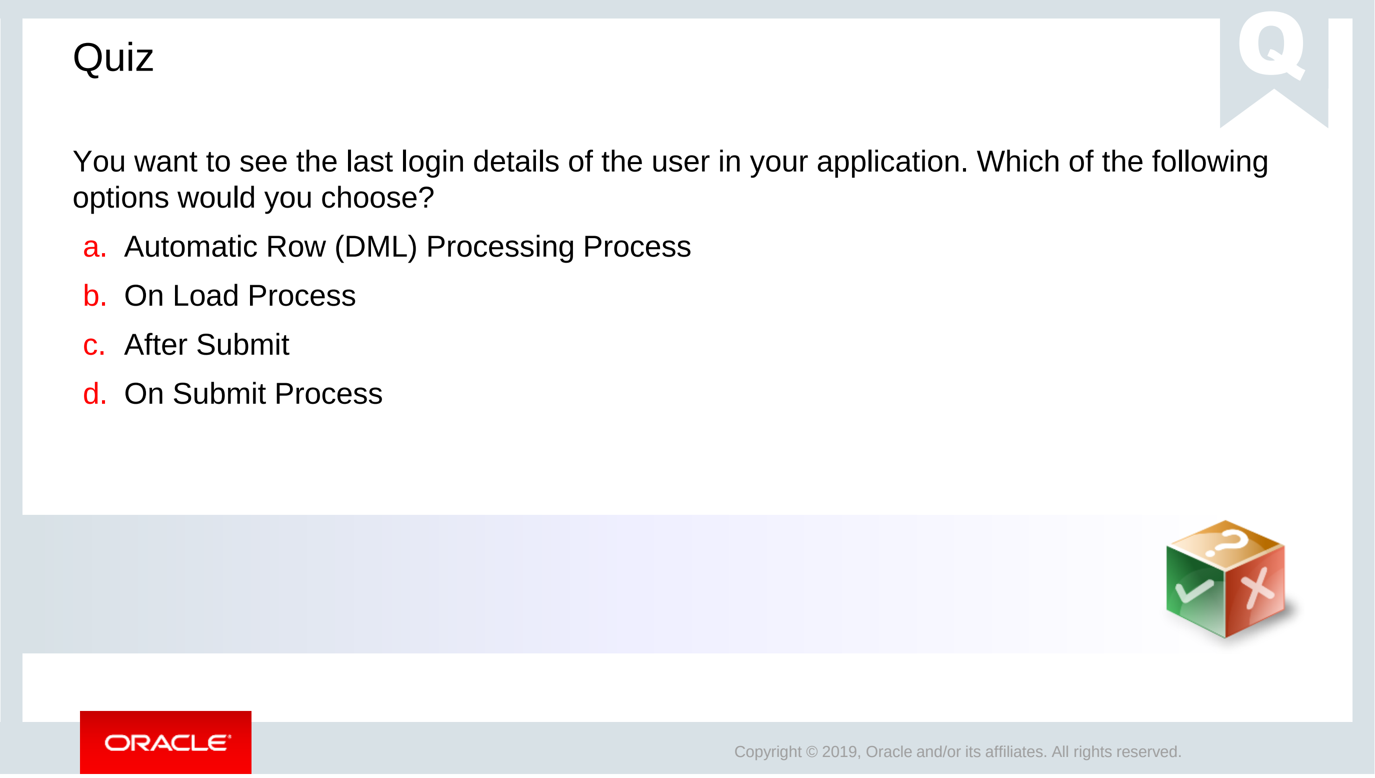
1. Navigate to the page definition for the page where you want to display the details. Here Steve selected *Home* page.
2. Create a display only item on Home page to display last login details:
   - Under **Rendering**, click **Page 1: Home.**
   - Click the **Items** button in the Gallery (located in the Grid Layout region of the page).
   - From the Items Gallery, drag **Display Only** item and drop it under **Items** in **Layout**. **Display Only** item is created.
   - Enter the following values in its Properties Pane on the right side:

     - **Identification: Name:** Enter `P1_LAST_LOGIN`

     - **Identification: Type:** Select **Display Only**

     - **Label: Label:** Enter `Last login`
3. To create On Load Process, expand **Pre-Rendering** and right-click **After Header** and select *Create Process* to create a new Process.

4. Enter the following values in the Properties Pane of the new process on the right side:
    - **Identification**: **Name**: Enter `P1_LAST_LOGIN_DETAILS`
    - **Identification**: **Type**: Select **PL/SQL Code**
    - **Source: PL/SQL Code** (code text area): Enter `DECLARE LOGDETAILS VARCHAR2(100); BEGIN SELECT MAX(LOGIN_DATE) INTO LOGDETAILS FROM LOGIN_TABLE GROUP BY LOGIN_USER HAVING LOGIN_USER = :app_user; :P1_LAST_LOGIN := LOGDETAILS; END;`
    - **Error: Error Message**: Enter `Error retrieving last login details!`
5. Click **Save and Run Page** icon to save and load the page. The process is created, and you can see *Last Login* time stamp displayed on the header of *Home* Page.

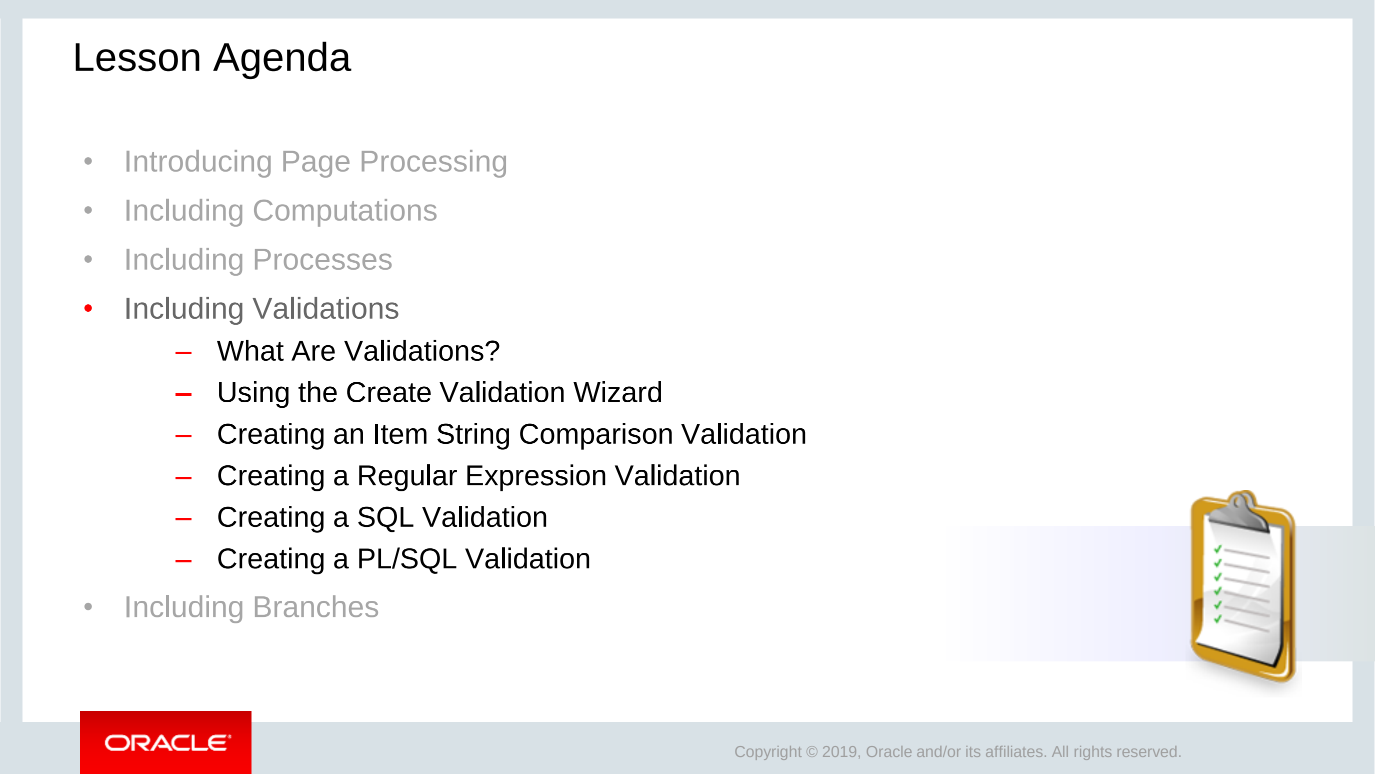


Quiz

You want to see the last login details of the user in your application. Which of the following options would you choose?

a. Automatic Row (DML) Processing Process
b. On Load Process
c. After Submit
d. On Submit Process

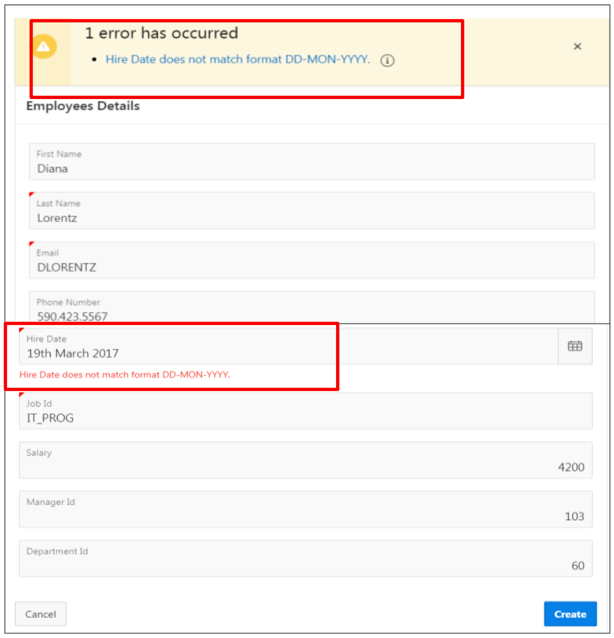
Answer: b



Lesson Agenda

- Introducing Page Processing
- Including Computations
- Including Processes
- Including Validations
  - What Are Validations?
  - Using the Create Validation Wizard
  - Creating an Item String Comparison Validation
  - Creating a Regular Expression Validation
  - Creating a SQL Validation
  - Creating a PL/SQL Validation
- Including Branches

What Are Validations?



1 error has occurred

- Hire Date does not match format DD-MON-YYYY.

Employees Details

First Name: Diana

Last Name: Lorentz

Email: DLORENTZ

Phone Number: 590.423.5567

Hire Date: 19th March 2017

Hire Date does not match format DD-MON-YYYY.

Job Id: IT_PROG

Salary: 4200

Manager Id: 103

Department Id: 60

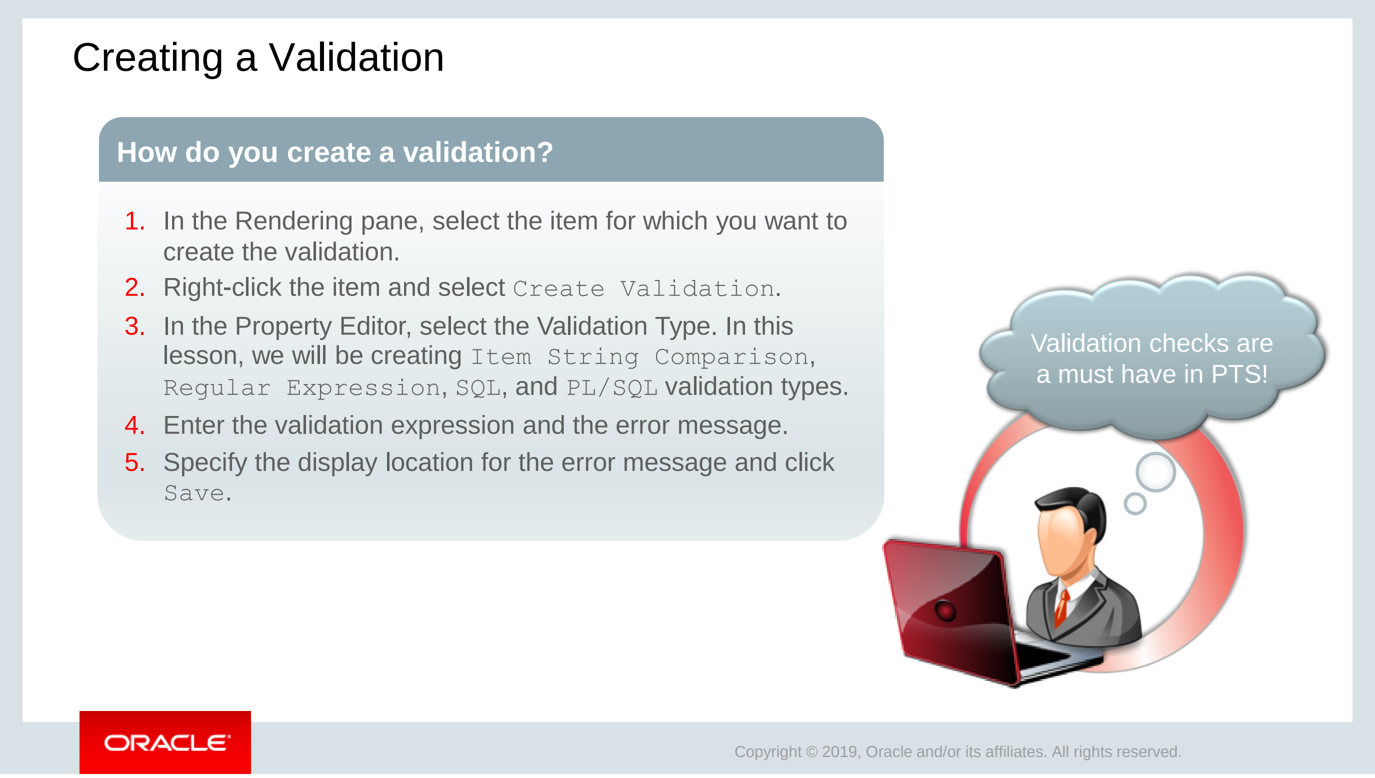
The slide example shows that you are getting an error message (in red) while creating an entry for a new employee. What could be the reason?

This form was created by using the *Create Form* Wizard, and as you see, here the *Hire Date* is a Date Picker item. This error message comes up due to a validation check in the back end as the date was not selected from the calendar icon next to the column.

When you use the *Create Form* Wizard, some validations are automatically created. For example, a *Not Null* validation is created for items that refer to a database column that is defined as *Not Null*. Similarly, if the database column is of type `NUMBER`, a validation to confirm that only numeric values are entered is created.

With Oracle Application Express, you can create validation checks to ensure data values are correct. You create validations to ensure that an application user enters valid and accurate data. If all the validations created on the page succeed, Oracle Application Express proceeds to the next step of processing; otherwise, Oracle Application Express redraws the page and displays the items along with the validation messages.

In the next few slides, you learn how to create validations.



Creating a Validation

How do you create a validation?

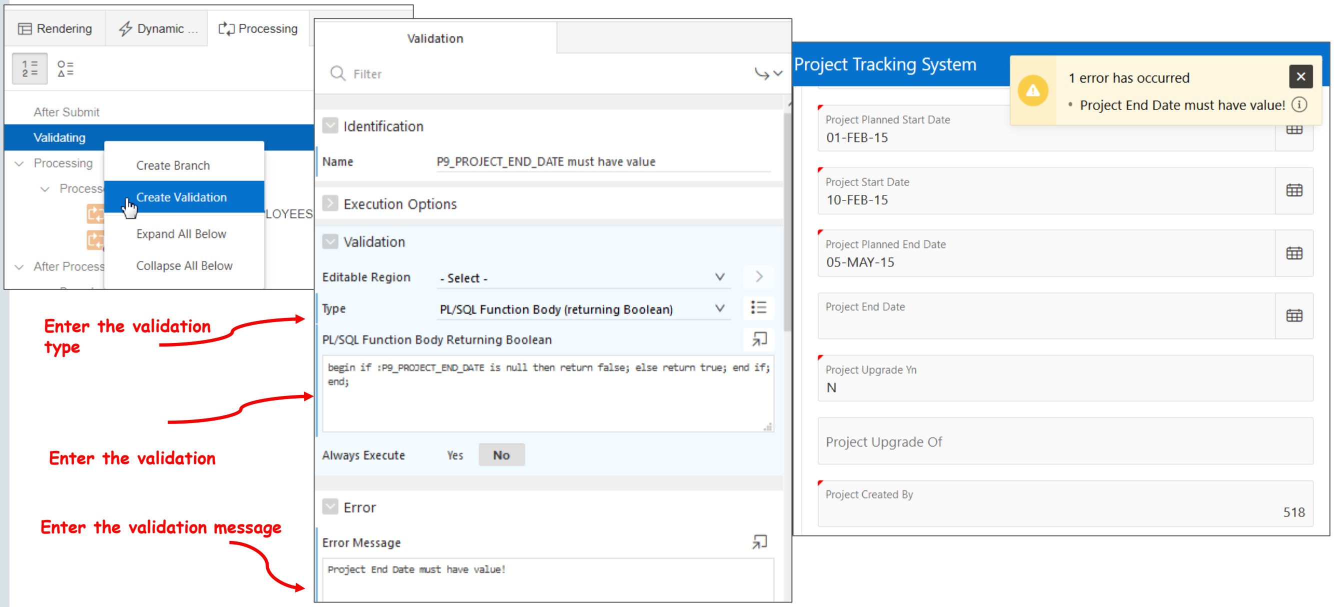
1. In the Rendering pane, select the item for which you want to create the validation.
2. Right-click the item and select `Create Validation`.
3. In the Property Editor, select the Validation Type. In this lesson, we will be creating `Item String Comparison`, `Regular Expression`, `SQL`, and `PL/SQL` validation types.
4. Enter the validation expression and the error message.
5. Specify the display location for the error message and click `Save`.

Validation checks are a must have in PTS!

As you know, to help the managers in his organization efficiently manage the existing and new projects, Steve has created multiple forms like *Manage Projects*, *Document Details*, *Create Employees,* and others. To prevent incorrect entries into the database, Steve now plans to add validations on some of the items in these forms.

This slide provides an overview of the steps to create a validation. Let’s see how he creates these validations in the next few slides.

Using the Create Validation Feature



Name: P9_PROJECT_END_DATE must have value

Editable Region: - Select -

Type: PL/SQL Function Body (returning Boolean)

PL/SQL Function Body Returning Boolean

```
begin if :P9_PROJECT_END_DATE is null then return false; else return true; end if;
end;
```

Always Execute: Yes [✓ No]

Error Message

```
Project End Date must have value!
```

Project Tracking System

1 error has occurred
- Project End Date must have value!

Project Planned Start Date: 01-FEB-15

Project Start Date: 10-FEB-15

Project Planned End Date: 05-MAY-15

Project End Date:

Project Upgrade Yn: N

Project Upgrade Of

Project Created By: 518

Enter the validation type

Enter the validation

Enter the validation message

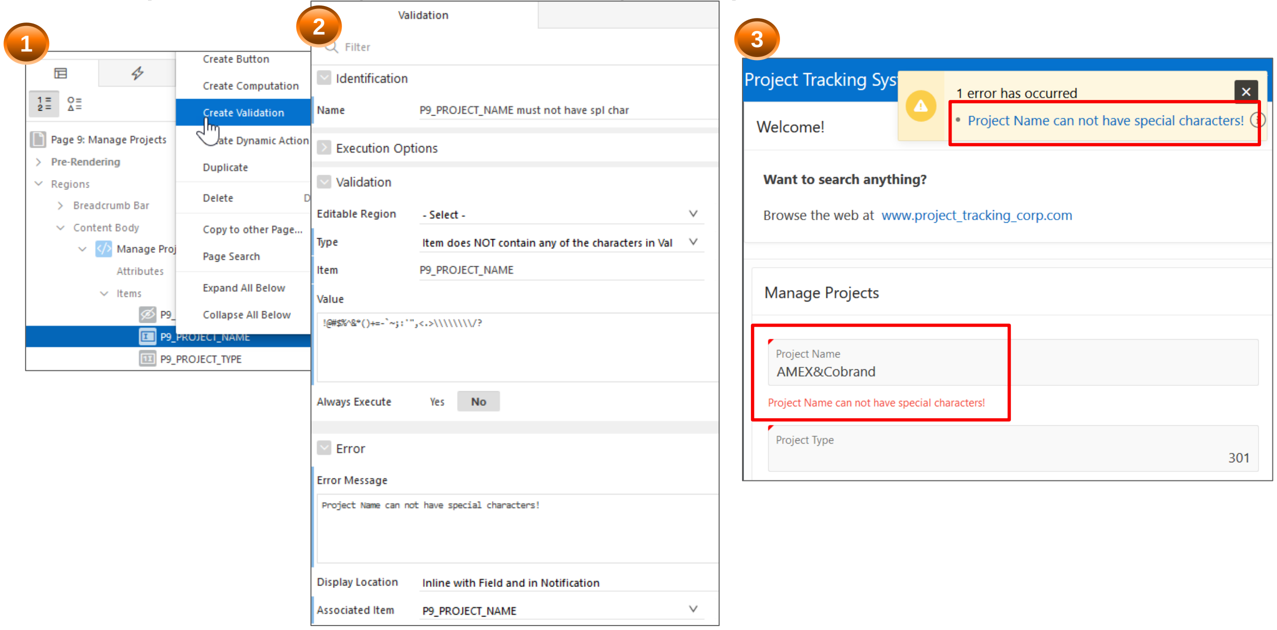
Steve here creates a validation that applies at record level in the table. In this example, he creates a validation on *PROJECTS* table such that if the *PROJECT_STATUS* is 103 or 104, then *PROJECT_END_DATE* field should have a value. Otherwise, an error message will be displayed.

Let's see how he creates a validation declaratively by using the **Create Validation** feature:

1. Navigate to the page definition of the page where you want to create the validation. Here Steve chooses **Manage Projects** form.
2. Click the **Processing** tab in the left pane.
3. Under **Processing**, right-click **Validations**; select *Create Validation* to create a new validation.
4. Enter the following values in the Properties Pane of the new validation on the right side:
   - **Identification: Name**: Enter `P9_PROJECT_END_DATE` *must have value*
   - **Validation: Type**: Select **PL/SQL Function Body (Returning Boolean)**
   - **Validation: PL/SQL Function Body (Returning Boolean)** (code text area): Enter `begin if :P9_PROJECT_END_DATE is null then return false; else return true; end if; end;`
   - **Error**: **Error Message**: Enter `Project End Date must have value!`
5. Click **Save**. The new validation is created.
6. Run the page and click **Cancel** to see *Projects Master Report*. Click edit icon beside any row to edit it. Leave **Project End Date** field blank and click **Apply Changes**. You see an error message displayed.

In the next few slides, you will learn how to create *Item String Comparison*, *Regular Expression*, *SQL*, and *PL/SQL* validations.

Example: Creating an Item String Comparison Validation



Identification

Name: P9_PROJECT_NAME must not have spl char

Execution Options

Validation

Editable Region: - Select -

Type: Item does NOT contain any of the characters in Val

Item: P9_PROJECT_NAME

Value:

```
!@#$%^&*()+=-`~;:'",<.>\\\\\\\\/?
```

Always Execute: Yes [✓ No]

Error

Error Message:

```
Project Name can not have special characters!
```

Display Location: Inline with Field and in Notification

Associated Item: P9_PROJECT_NAME

Project Tracking Sys

1 error has occurred

- Project Name can not have special characters!

Welcome!

Want to search anything?

Browse the web at www.project_tracking_corp.com

Manage Projects

Project Name: AMEX&Cobrand

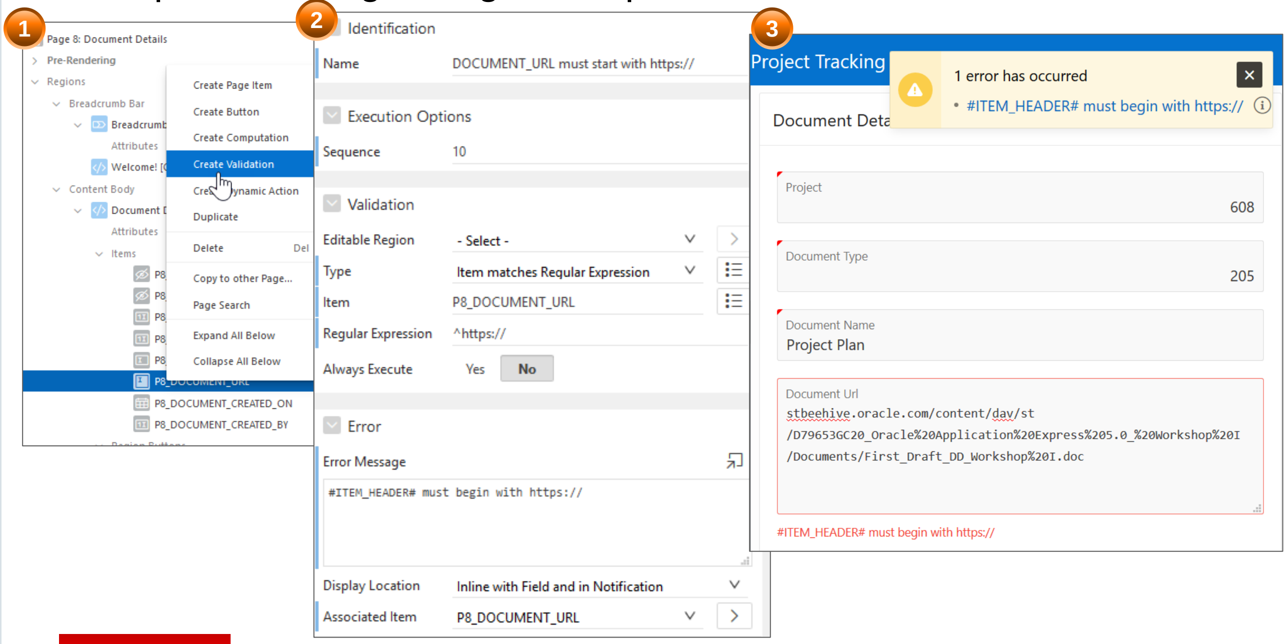
Project Name can not have special characters!

Project Type: 301

In this slide, Steve continues working on the *Manage Projects* form page. And this time, he wants to create a validation such that the *PROJECT_NAME* does not have any special characters. He wants the *Project Name* to be entered in the correct format before it gets inserted or updated into the *PROJECTS* table. Let's now see how he creates this validation by using *Item String Comparison* validation type:

1. Open **Page 9: Manage Projects** in Page Designer view.
2. Under Rendering, right-click **P9_PROJECT_NAME** and select *Create Validation*.
3. Enter the following values in the Properties Pane of the new validation on the right side:
   - **Identification**: **Name**: Enter `P9_PROJECT_NAME must not have spl char`
   - **Validation**: **Type**: Select **Item does NOT contain any of the characters in value**
   - **Validation**: **Item**: Select (from Popup LOV) **`P9_PROJECT_NAME`**
   - **Validation: Value** (code text area): Enter `!@#$%^&*()+=-`~;:'",<.>\?`
   - **Error: Error Message**: Enter `Project Name can not have special characters!`
4. Click Save. The new validation is created.
5. Run the page and choose any existing row and click the edit icon. Modify the project name as *`abd@123`* and click **Apply Changes**. You see an error message displayed.

Example: Creating a Regular Expression Validation



Identification

Name	DOCUMENT_URL must start with https://

Execution Options

Sequence	10

Validation

Editable Region	- Select -
Type	Item matches Regular Expression
Item	P8_DOCUMENT_URL
Regular Expression	^https://
Always Execute	Yes **No**

Error

Error Message

```
#ITEM_HEADER# must begin with https://
```

Display Location	Inline with Field and in Notification
Associated Item	P8_DOCUMENT_URL

1 error has occurred

- #ITEM_HEADER# must begin with https://

Project	608
Document Type	205
Document Name	Project Plan
Document Url	stbeehive.oracle.com/content/dav/st/D79653GC20_Oracle%20Application%20Express%205.0_%20Workshop%20I/Documents/First_Draft_DD_Workshop%20I.doc

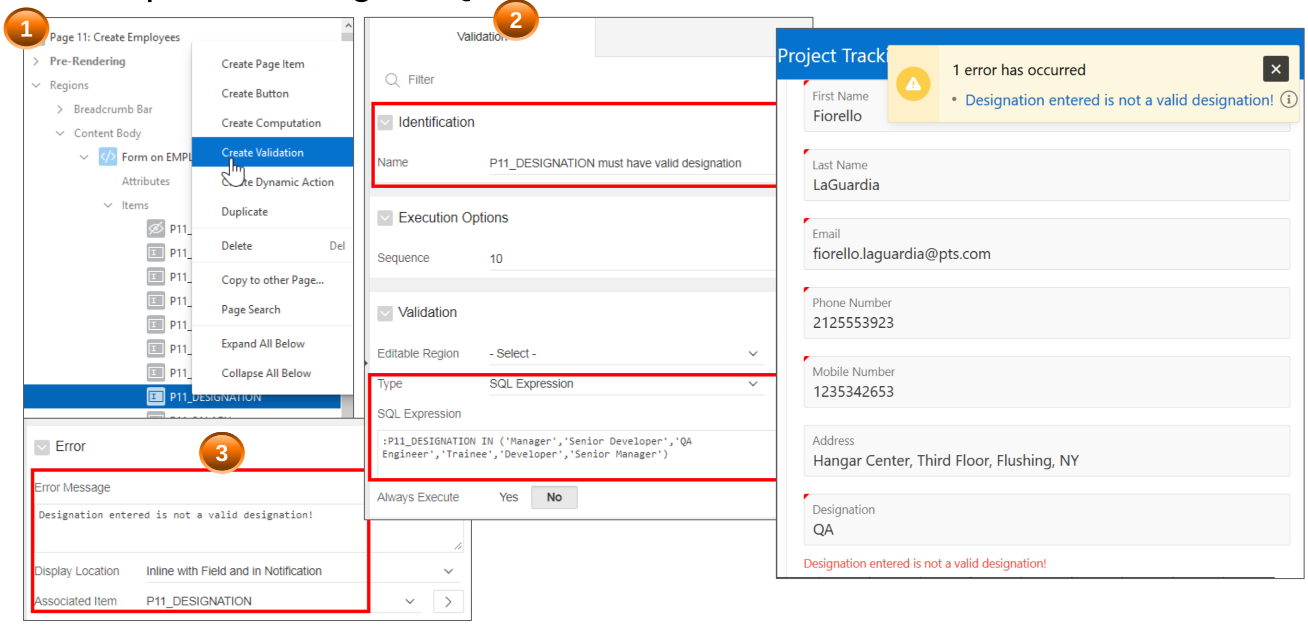
#ITEM_HEADER# must begin with https://

Steve created the *Projects Master Document* report and *Document Details* form when he was creating the interactive grid. Now he wants to add some validations to the *Document Details* form page. He wants the Project Document URL to be entered in the correct format before it gets inserted/updated into the *PROJECT_DOCUMENTS* table. And for that he wants to create a validation on DOCUMENT_URL, such that the Document URL should start with *https://* .

Let's now see how he creates this validation by using *Regular Expression* validation type:

1. Open **Page 8: Documents Detail** in Page Designer view.
2. Under **Rendering**, right-click **P8_DOCUMENT_URL** (under DOCUMENTS Detail) and select **Create Validation**.
3. Enter the following values in the Properties Pane of the new validation on the right side:
   - **Identification: Name**: Enter `DOCUMENT_URL must start with https://`
   - **Validation**: **Type**: Select **Item matches Regular Expression**
   - **Validation**: **Item**: Select (from Popup LOV) **P8_DOCUMENT_URL**
   - **Regular Expression** (code text area): Enter `^https://`
   - **Error: Error Message**: Enter `#ITEM_HEADER# must begin with https://`
4. Click **Save**. The new validation is created.
5. Run the page and click **Cancel** to see list of document types. Now, click edit icon beside any row and try to modify the *Document URL* field of first row to remove `^https://` or adding *`www://`*. Click **Save** and you get an error message.

Example: Creating a SQL Validation



Identification

Name: P11_DESIGNATION must have valid designation

Execution Options

Sequence: 10

Validation

Editable Region: - Select -

Type: SQL Expression

SQL Expression:

```
:P11_DESIGNATION IN ('Manager','Senior Developer','QA
Engineer','Trainee','Developer','Senior Manager')
```

Always Execute: Yes [✓ No]

Error

Error Message:

```
Designation entered is not a valid designation!
```

Display Location: Inline with Field and in Notification

Associated Item: P11_DESIGNATION

Project Track

1 error has occurred

- Designation entered is not a valid designation!

First Name: Fiorello

Last Name: LaGuardia

Email: fiorello.laguardia@pts.com

Phone Number: 2125553923

Mobile Number: 1235342653

Address: Hangar Center, Third Floor, Flushing, NY

Designation: QA

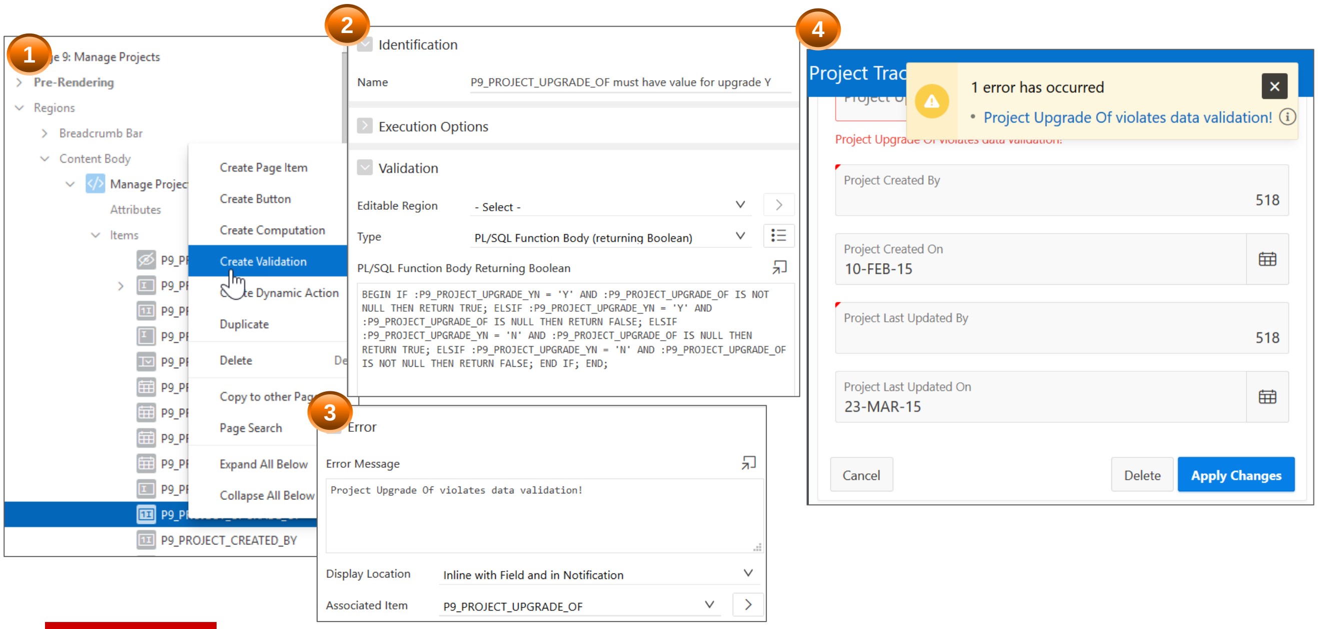
Designation entered is not a valid designation!

Now, Steve wants to add some validations in his *Create Employees* form. Every time a new employee joins in, his/her details are added into this form, which later gets updated into the employees database.

Therefore, Steve does not want any invalid/incorrect designations entered into the *Designation* column. Let's see how he creates a SQL validation to do so:

1. Open **Create Employees** in Page Designer view.
2. Under **Rendering**, right-click **P11_DESIGNATION** (under Page Items) and select **Create Validation**.
3. Enter the following values in the Properties Pane of the new validation on the right side:
   - **Identification: Name**: Enter `P11_DESIGNATION must have valid designation`
   - **Validation: Type**: Select *SQL Expression*
   - **Validation: SQL Expression** (code text area): Enter `:P11_DESIGNATION IN ('Manager','Senior Developer','QA Engineer','Trainee','Developer','Senior Manager')`
   - **Error: Error Message**: Enter `Designation entered is not a valid designation!`
4. Click **Save**. The new validation is created.
5. Run the page. Now, try entering values for a new record, giving designation as *`Singer`*, and click **Create**. You should get an error message in the notification area and next to the **Designation** item.

Example: Creating a PL/SQL Validation



Name: P9_PROJECT_UPGRADE_OF must have value for upgrade Y

Editable Region: - Select -

Type: PL/SQL Function Body (returning Boolean)

PL/SQL Function Body Returning Boolean

```
BEGIN IF :P9_PROJECT_UPGRADE_YN = 'Y' AND :P9_PROJECT_UPGRADE_OF IS NOT
NULL THEN RETURN TRUE; ELSIF :P9_PROJECT_UPGRADE_YN = 'Y' AND
:P9_PROJECT_UPGRADE_OF IS NULL THEN RETURN FALSE; ELSIF
:P9_PROJECT_UPGRADE_YN = 'N' AND :P9_PROJECT_UPGRADE_OF IS NULL THEN
RETURN TRUE; ELSIF :P9_PROJECT_UPGRADE_YN = 'N' AND :P9_PROJECT_UPGRADE_OF
IS NOT NULL THEN RETURN FALSE; END IF; END;
```

Error Message: Project Upgrade Of violates data validation!

Display Location: Inline with Field and in Notification

Associated Item: P9_PROJECT_UPGRADE_OF

1 error has occurred

- Project Upgrade Of violates data validation!

Project Upgrade Of violates data validation!

Project Created By: 518

Project Created On: 10-FEB-15

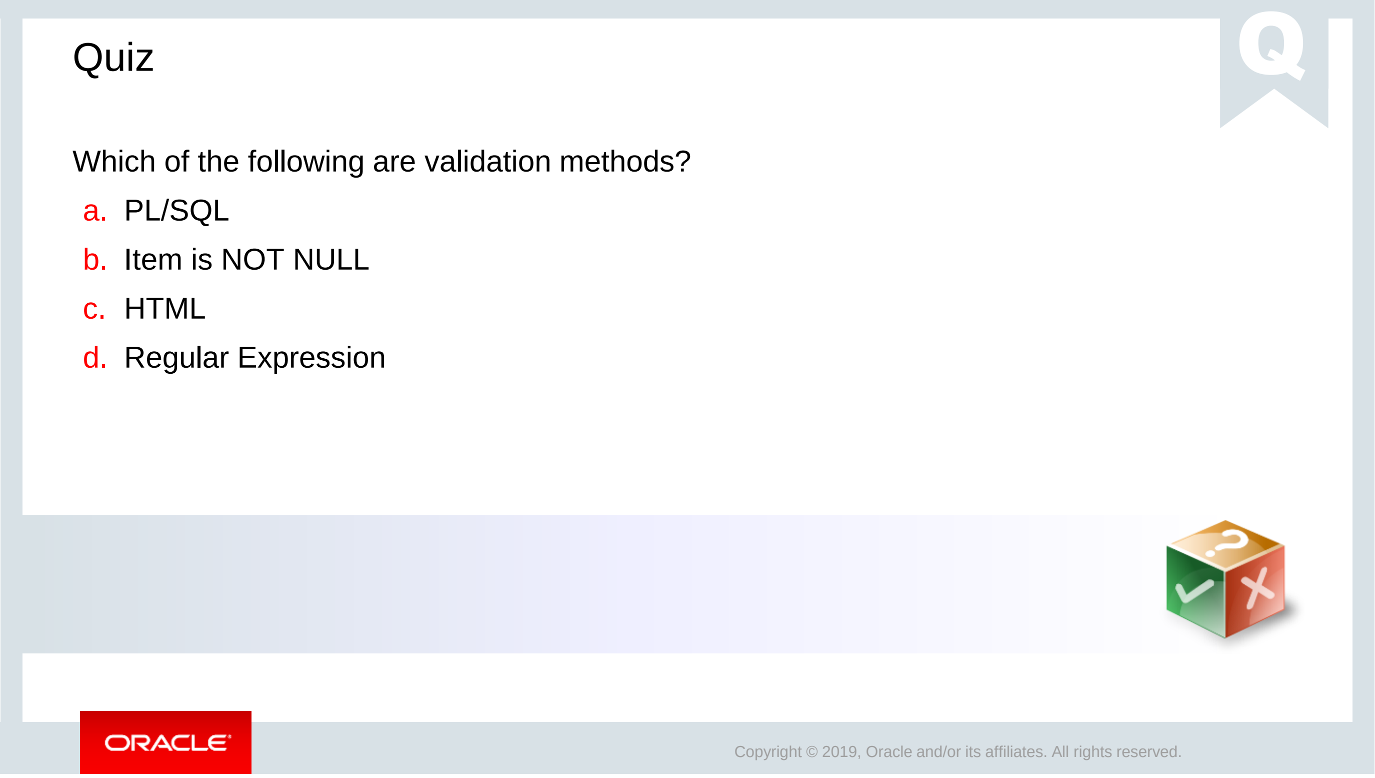
Project Last Updated By: 518

Project Last Updated On: 23-MAR-15

Steve further enhances his *Manage Projects* form page by adding another validation type. He creates a validation to check if **Project Upgrade Of** field is populated when **Project Upgrade YN** item type is set to `Y(Yes)` and does not populate any value when **Project Upgrade YN** item type is set to `N(No)`. Let's see how he uses the validation type *PL/SQL Function Body* to do so:

1. Open Manage Projects in Page Designer view.
2. Under Rendering, right-click P9_PROJECT_UPGRADE_OF (under Page Items) and select *Create Validation*.
3. Enter the following values in the Properties Pane of the new validation on the right side:
   - **Name**: Enter `P9_PROJECT_UPGRADE_OF must have value for upgrade Y`
   - **Validation: Type**: Select **PL/SQL function body returning Boolean**
   - **Validation: PL/SQL Function Body Returning Boolean** (code text area): Enter

     ```
     BEGIN IF :P9_PROJECT_UPGRADE_YN = 'Y' AND
     :P9_PROJECT_UPGRADE_OF IS NOT NULL THEN RETURN TRUE; ELSIF
     :P9_PROJECT_UPGRADE_YN = 'Y' AND :P9_PROJECT_UPGRADE_OF IS NULL
     THEN RETURN FALSE; ELSIF :P9_PROJECT_UPGRADE_YN = 'N' AND
     :P9_PROJECT_UPGRADE_OF IS NULL THEN RETURN TRUE; ELSIF
     :P9_PROJECT_UPGRADE_YN = 'N' AND :P9_PROJECT_UPGRADE_OF IS NOT
     NULL THEN RETURN FALSE; END IF; END;
     ```
   - **Error: Error Message**: Enter `Project Upgrade Of violates data validation!`
4. Click Save. The new validation is created.
5. Run the page. Select *Y* for Project Upgrade YN and leave the Project Upgrade Of field blank and click Apply Changes. You should get an error message next to the Project Upgrade Of item.



Quiz

Which of the following are validation methods?

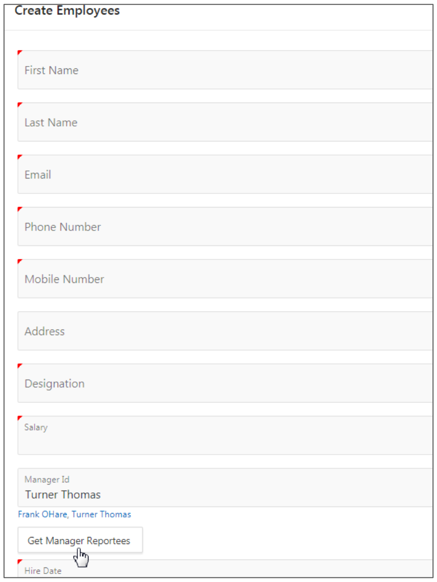
a. PL/SQL
b. Item is NOT NULL
c. HTML
d. Regular Expression

Answer: a, b, and d

Lesson Agenda

- Introducing Page Processing
- Including Computations
- Including Processes
- Including Validations
- Including Branches
  - What Is Branching?
  - Creating a Branch

What Is Branching?



Create Employees

First Name

Last Name

Email

Phone Number

Mobile Number

Address

Designation

Salary

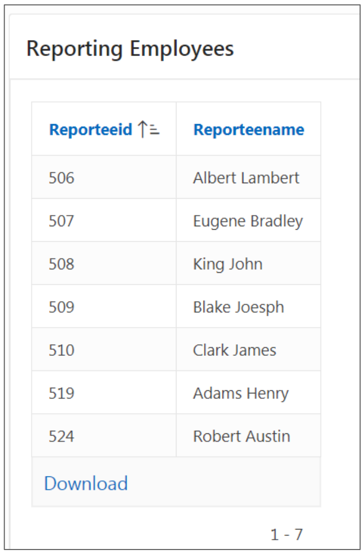
Manager Id: Turner Thomas

Frank OHare, Turner Thomas

Get Manager Reportees

Hire Date

The page is submitted and is redirected to a new page



Reporting Employees

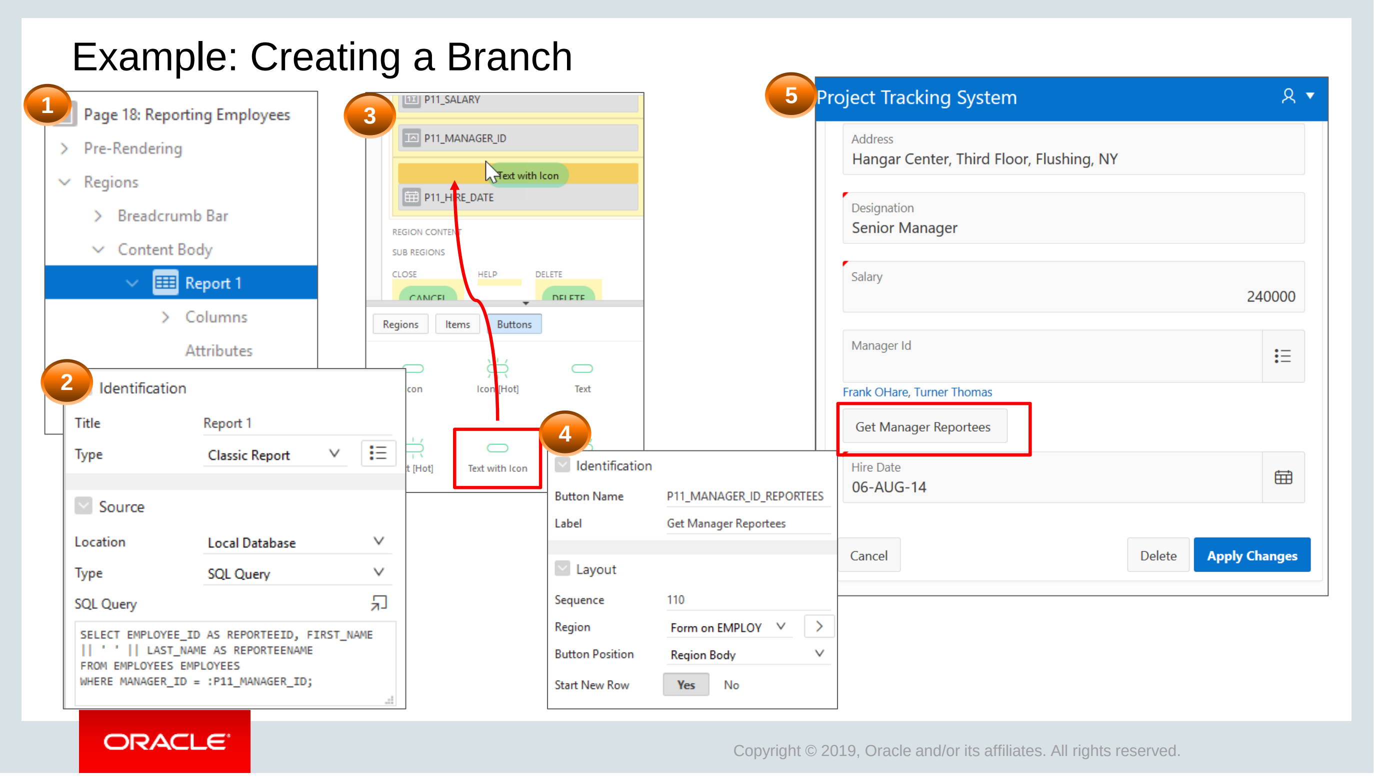
Reporteeid	Reporteename
506	Albert Lambert
507	Eugene Bradley
508	King John
509	Blake Joesph
510	Clark James
519	Adams Henry
524	Robert Austin

Download

1 - 7

In the above example, when you select a particular manager in the *Create Employees* form, and then click the *Get Manager Reportees* button, the *Create Employees* form page gets submitted, and you are redirected to a report displaying all the employees reporting to that manager. How do you think it happens?

And this is where branching comes in. A branch links to another page in your application or to a URL. The Application Express engine considers branching at different times during page processing. You can choose to branch before processing, before computation, before validation, and after processing. Similar to the example in the above slide, you have a form page that accepts values from the user. After the form page is submitted, the Oracle Application Express engine executes the branch that navigates the user to another page. If the *Cancel* button is clicked, no processing occurs, and the user is redirected to another page. In the next slide, Steve creates a new branch when a *Page Submit* event occurs.



Example: Creating a Branch

1 Page 18: Reporting Employees
- Pre-Rendering
- Regions
  - Breadcrumb Bar
  - Content Body
    - Report 1
      - Columns
      - Attributes

2 Identification

Title	Report 1
Type	Classic Report

Source

Location	Local Database
Type	SQL Query
SQL Query	

```
SELECT EMPLOYEE_ID AS REPORTEEID, FIRST_NAME
|| ' ' || LAST_NAME AS REPORTEENAME
FROM EMPLOYEES EMPLOYEES
WHERE MANAGER_ID = :P11_MANAGER_ID;
```

4 Identification

Button Name	P11_MANAGER_ID_REPORTEES
Label	Get Manager Reportees

Layout

Sequence	110
Region	Form on EMPLOY
Button Position	Region Body
Start New Row	Yes No

5 Project Tracking System

Address	Hangar Center, Third Floor, Flushing, NY
Designation	Senior Manager
Salary	240000
Manager Id	

Frank OHare, Turner Thomas

Get Manager Reportees

Hire Date	06-AUG-14

As you learned in the previous slide, Steve now wants to create a branch when a *Page Submit* event occurs. He wants to add a new *Button* item to the *Create Employees* form, and on the event of the button click, branch to another page with list of employees reporting to the manager name selected on the form. Let's see how he does it by following the steps below:

1. Steve first creates a classic report (you learned how to create it in the lesson on *Creating Classic Reports*) with a list of employees reporting to a particular manager. Enter the following values:
   - **Page Number**: Enter `18`
   - **Page Name**: Enter `Reporting Employees`
   - **Navigation Preference**: Accept defaults, that is, select **Do not associate this page with a navigation menu entry**
   - **Report Source:** Select **SQL Query.**
   - **SQL Query** : Paste the below query in the SQL Query editor (screenshot 2):

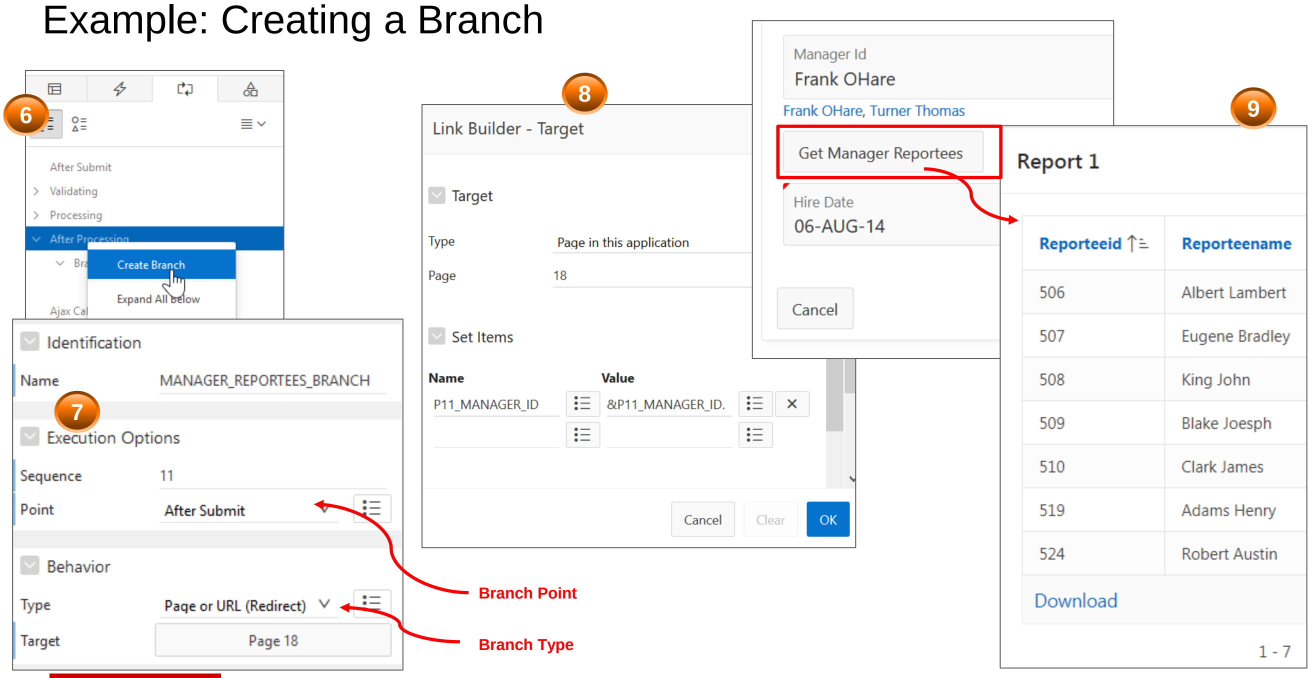
```
SELECT EMPLOYEE_ID AS REPORTEEID, FIRST_NAME || ' ' ||
LAST_NAME AS REPORTEENAME
FROM EMPLOYEES EMPLOYEES
WHERE MANAGER_ID = :P11_MANAGER_ID;
```

The classic report page *Reporting Employees* will be displayed in Page Designer view.

2. Steve now adds a button (you learned how to create a button in a previous lesson on *Items and Buttons*) under `P11_MANAGER_ID` on **Create Employees** form (screenshot 3). In the *Create Employees* form, he enters the following values and creates and saves a button (screenshot 4).
    - **Identification: Button Name**: Enter `P11_MANAGER_ID_REPORTEES`
    - **Identification: Label**: Enter `Get Manager Reportees`
    - **Behavior: Action**: Select **Submit Page**

   You can see the new button *Get Manager Reportees* below **Manager ID** on the Create Employees form.

Example: Creating a Branch



Identification
Name: MANAGER_REPORTEES_BRANCH

Execution Options
Sequence: 11
Point: After Submit

Behavior
Type: Page or URL (Redirect)
Target: Page 18

Link Builder - Target

Target
Type: Page in this application
Page: 18

Set Items

Name	Value
P11_MANAGER_ID	&P11_MANAGER_ID.

Manager Id
Frank OHare

Frank OHare, Turner Thomas

Get Manager Reportees

Hire Date
06-AUG-14

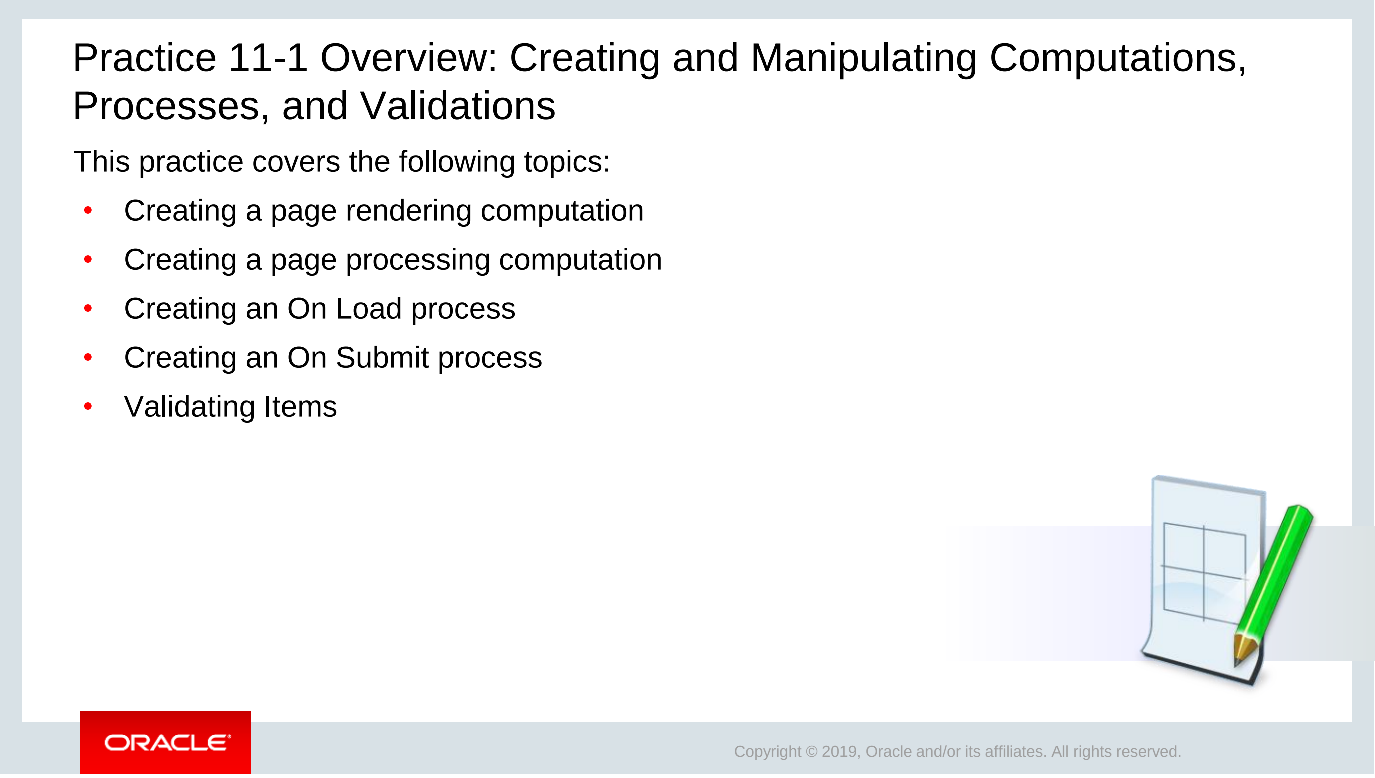
Cancel

Report 1

Reporteeid	Reporteename
506	Albert Lambert
507	Eugene Bradley
508	King John
509	Blake Joesph
510	Clark James
519	Adams Henry
524	Robert Austin

1 - 7

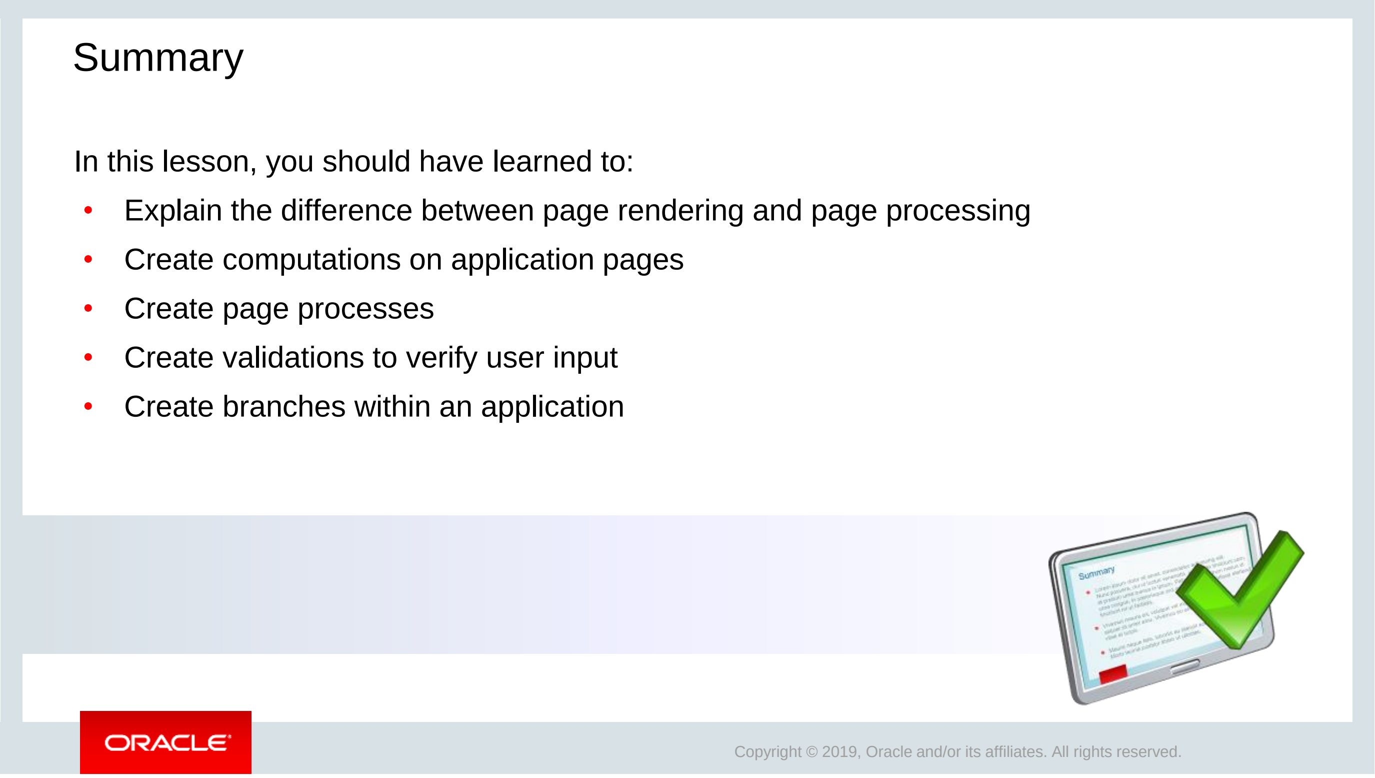
3. Now, Steve creates a branch to *Page 18: Reporting Employees* on submit of the button created on the *Create Employees* form:
   - Open **Create Employees** form in Page Designer.
   - Under **Processing**, right click **Branches** and select *Create Branch* (screenshot 6).
   - Enter the following values in the Properties Pane of the new branch on the right side (screenshot 7 and 8):
     - **Identification: Name**: Enter `MANAGER_REPORTEES_BRANCH`
     - **Execution Options: Point**: Select **After Submit**
     - **Behavior: Type**: Click **Page or URL (Redirect)**
     - **Behavior: Target**: Select `Page 18` from the popup LOV
     - **Behavior > Set Items:** For **Name,** select `P11_MANAGER_ID`, for **Value** select `&P11_MANAGER_ID`, and click **OK**
   - Click the **Save and Run Page** icon to load the page.
   - Select a manager and click the button *Get Manager Reportees* below *Manager ID*. The *Reporting Employees* report will get loaded with a list of reportees of the selected manager.



Practice 11-1 Overview: Creating and Manipulating Computations, Processes, and Validations

This practice covers the following topics:

- Creating a page rendering computation
- Creating a page processing computation
- Creating an On Load process
- Creating an On Submit process
- Validating Items



Summary

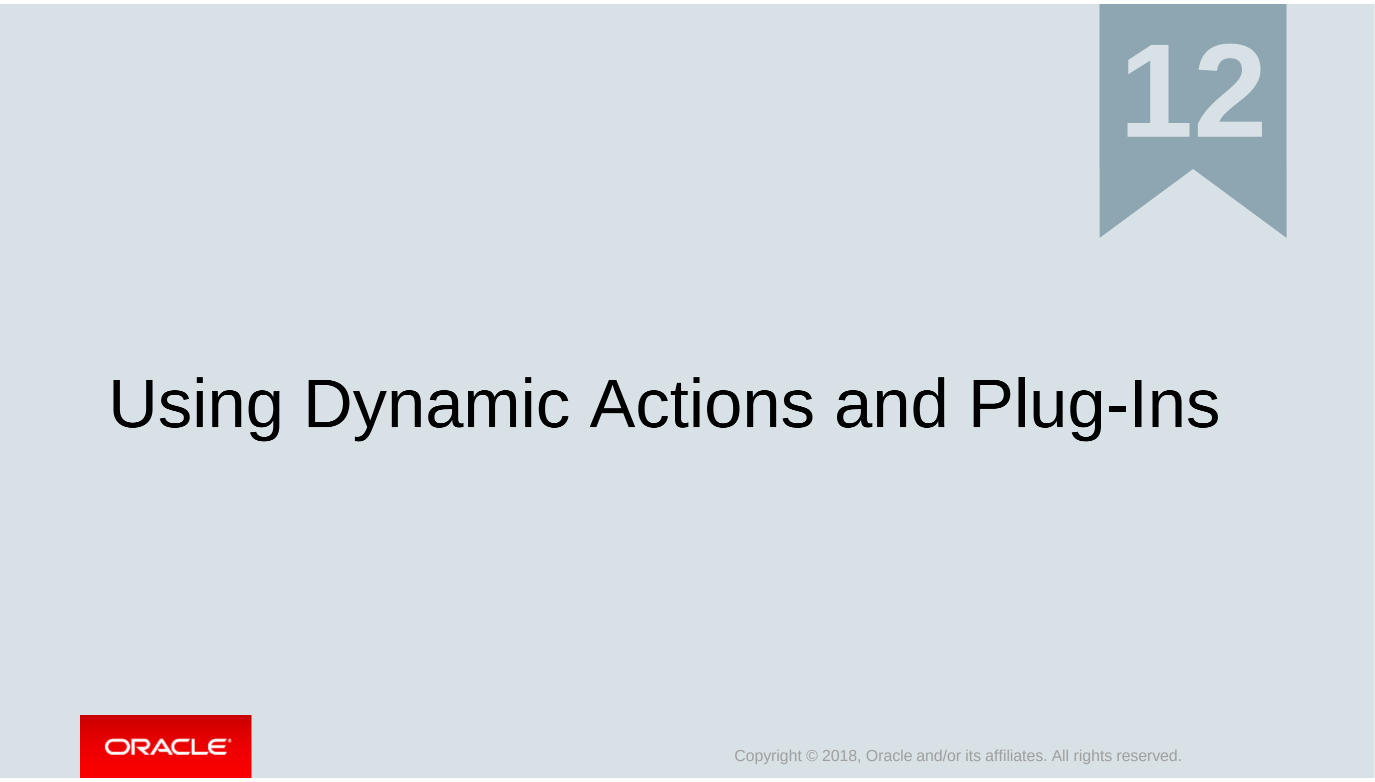
In this lesson, you should have learned to:

- Explain the difference between page rendering and page processing
- Create computations on application pages
- Create page processes
- Create validations to verify user input
- Create branches within an application

Summary

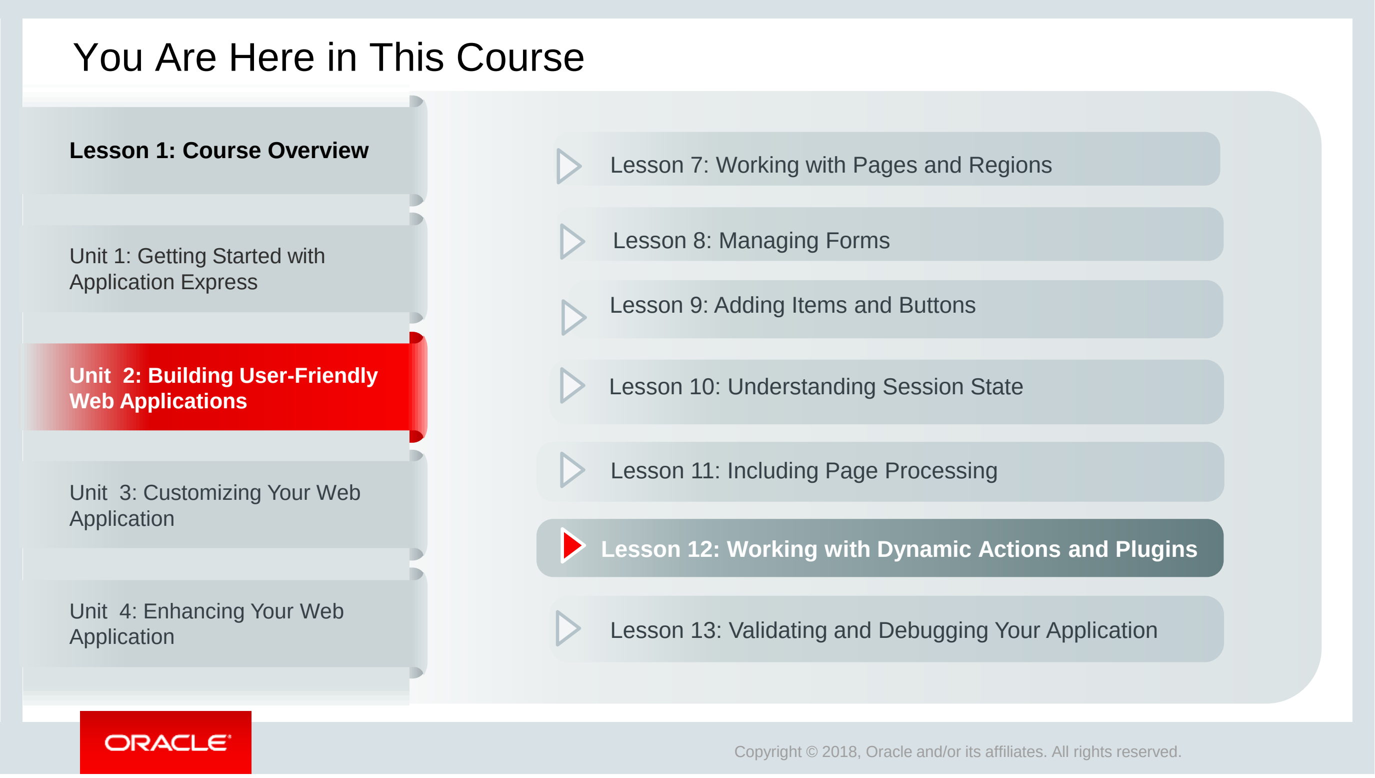
ORACLE

This lesson explained the difference between page rendering and page processing. You should also have learned how to create computations, processes, and validations.



12

Using Dynamic Actions and Plug-Ins



You Are Here in This Course

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

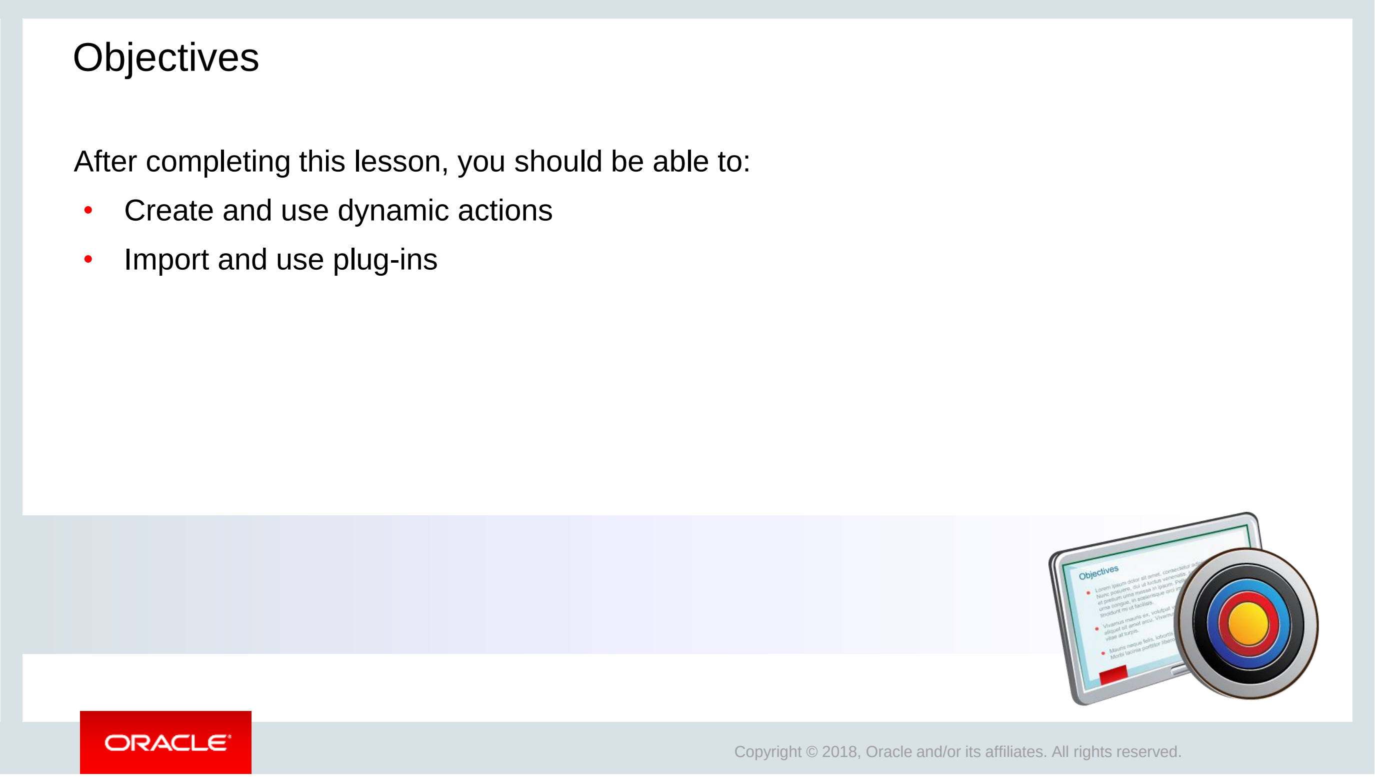
Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- Lesson 7: Working with Pages and Regions
- Lesson 8: Managing Forms
- Lesson 9: Adding Items and Buttons
- Lesson 10: Understanding Session State
- Lesson 11: Including Page Processing
- **Lesson 12: Working with Dynamic Actions and Plugins**
- Lesson 13: Validating and Debugging Your Application

This slide is a graphical depiction of the course, highlighting Unit 2 - Lesson 12 in particular, which is dealt with in these slides.



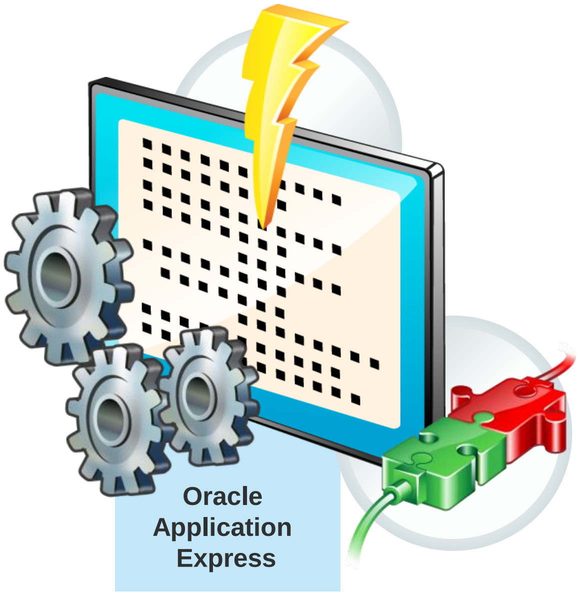
Objectives

After completing this lesson, you should be able to:

- Create and use dynamic actions
- Import and use plug-ins

This lesson discusses dynamic actions and plug-ins. You learn what they are, how to create dynamic actions, how to import and install plug-ins, and how to use it in your application.

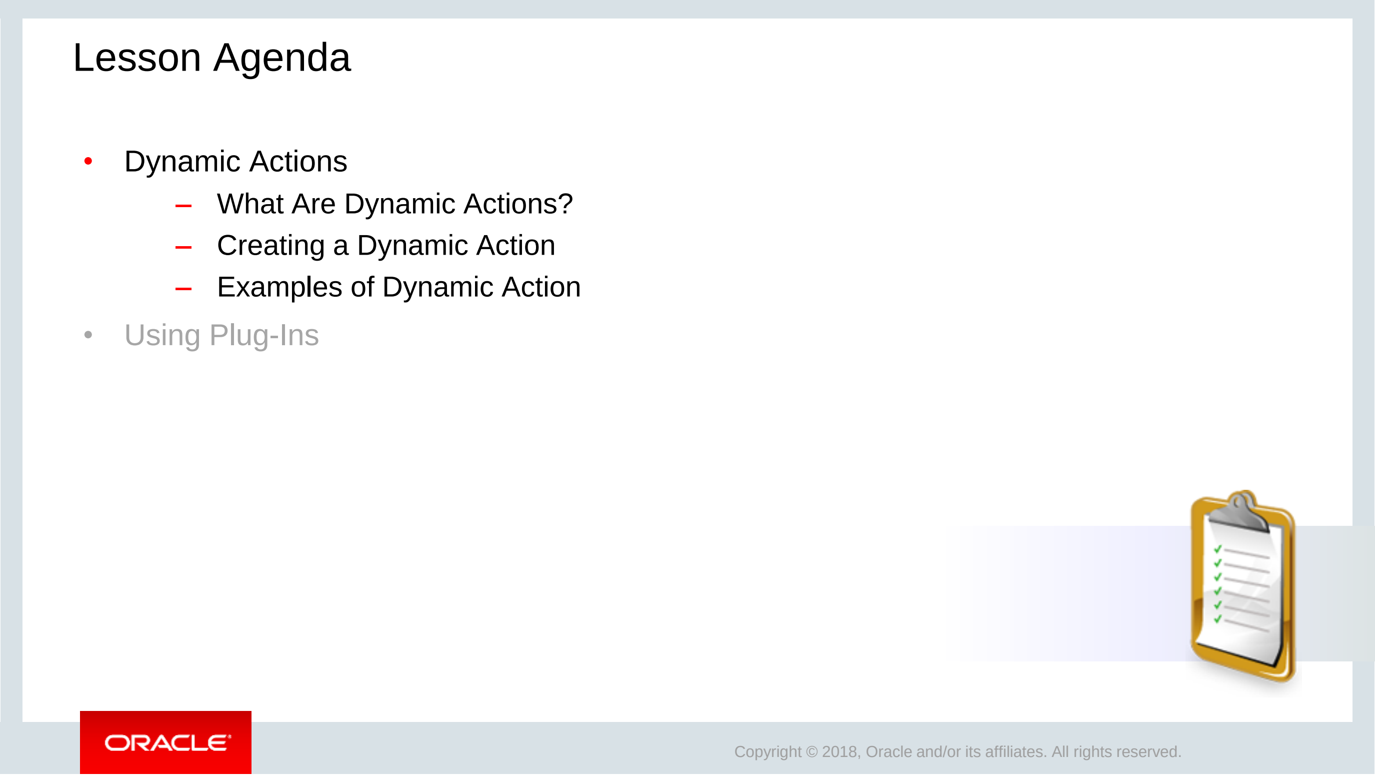
Steve Uses Dynamic Actions and Plug-ins



After having created the PTS application, along with the reports, forms, and pages in the application, Steve is now looking at enhancing the user experience of the application. He comes across the Oracle Application Express built-in Dynamic Actions and Plug-ins features, which helps in improving the user experience.

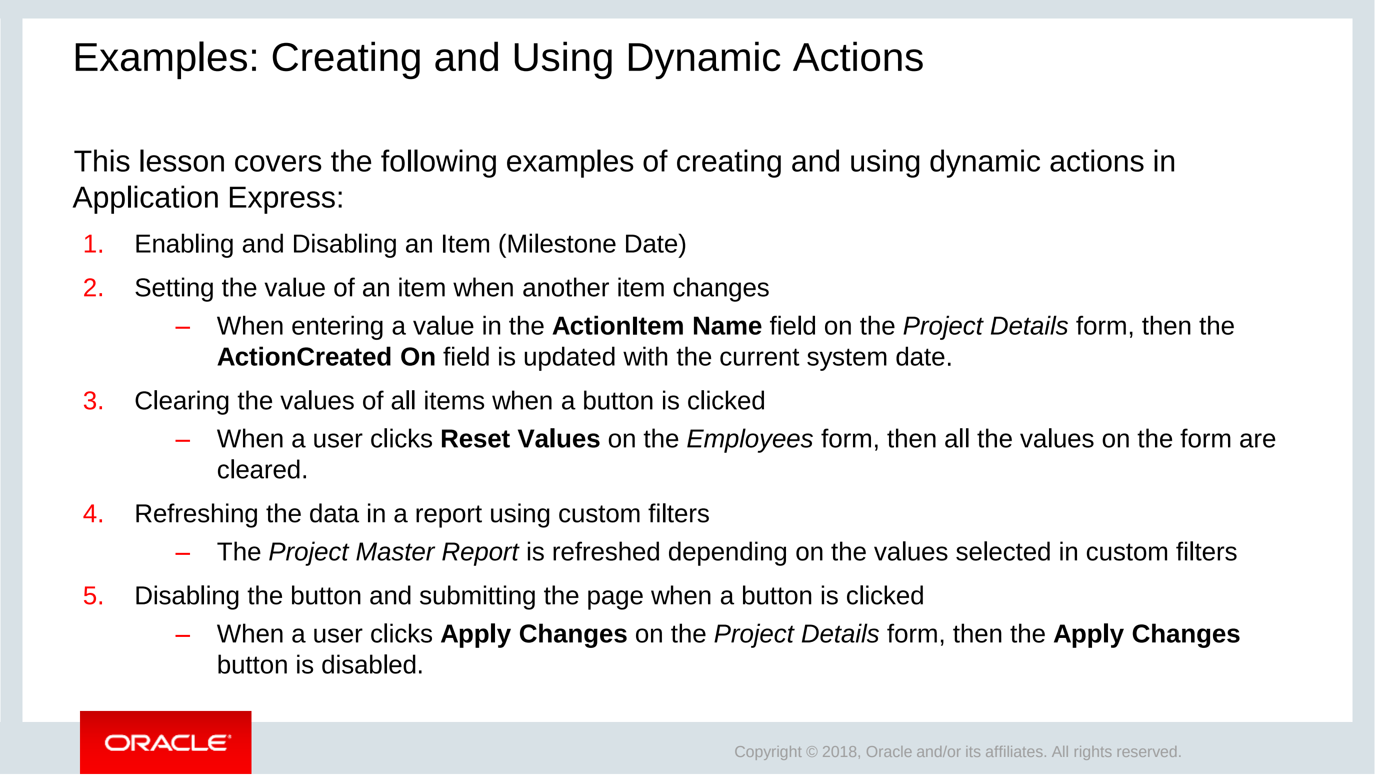
He figures out a couple of areas that he considers apt to implement dynamic actions, thereby enhancing user experience as well as the usability of the product. Some of these areas are:

- Enabling and disabling the **Milestone Date** field on the *Project Members* form based on certain conditions
- When entering a value in the **ActionItem Name** field on the *Project_ActionItem* form, the **ActionCreated On** field should get updated with the current system date.
- Clearing the values of all items in a form when clicking the **Reset** button. He considers implementing dynamic action on the **Reset** button with the label **Reset Values** on the *Employees* form so that on clicking **Reset Values** all the preset values on the form are cleared.
- Disabling the **Apply Changes** button on the *Project Details* form when a user clicks the button. When the **Apply Changes** button is clicked, page processing is triggered at the back end. But sometimes, it so happens that the user clicks the button more than once, thereby creating a possibility of data integrity issue. Steve thinks it is an appropriate scenario to implement dynamic action on the **Apply Changes** button so that the button is disabled after the user clicks it once. This would mitigate all possible risks related to data integrity.
- To enhance the usability of the *Project Master Report* in the application, Steve considers adding custom filters in the report and implementing dynamic actions on those filters so that depending on the filter selection, the report is refreshed only with the relevant values.



Lesson Agenda

- Dynamic Actions
  - What Are Dynamic Actions?
  - Creating a Dynamic Action
  - Examples of Dynamic Action
- Using Plug-Ins



Examples: Creating and Using Dynamic Actions

This lesson covers the following examples of creating and using dynamic actions in Application Express:

1. Enabling and Disabling an Item (Milestone Date)
2. Setting the value of an item when another item changes
   - When entering a value in the **ActionItem Name** field on the *Project Details* form, then the **ActionCreated On** field is updated with the current system date.
3. Clearing the values of all items when a button is clicked
   - When a user clicks **Reset Values** on the *Employees* form, then all the values on the form are cleared.
4. Refreshing the data in a report using custom filters
   - The *Project Master Report* is refreshed depending on the values selected in custom filters
5. Disabling the button and submitting the page when a button is clicked
   - When a user clicks **Apply Changes** on the *Project Details* form, then the **Apply Changes** button is disabled.

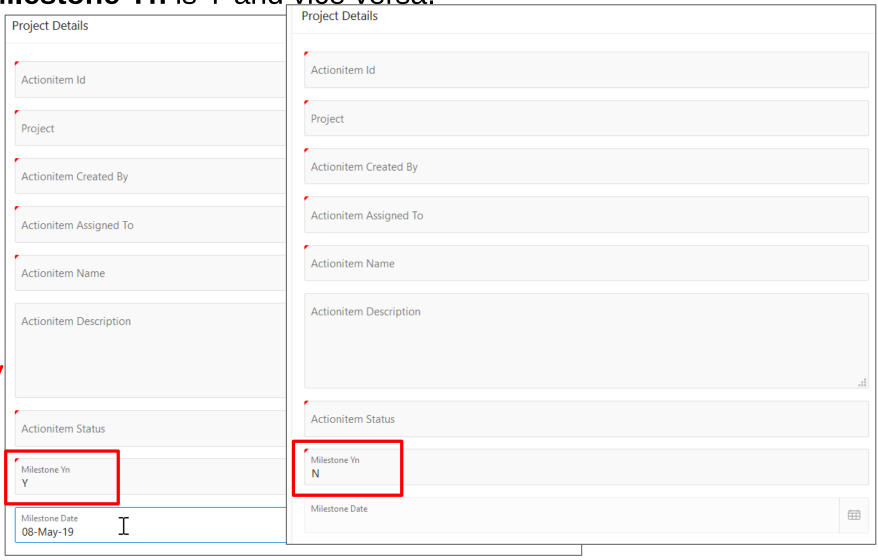
ORACLE

In this lesson, you examine some more dynamic action examples as listed in the slide.

What Is a Dynamic Action?

This is an example of a dynamic action that enables the **Milestone Date** field when the value in the **Milestone Yn** is Y and vice versa.

Milestone Date is enabled when action item is a milestone, that is when Milestone Yn = Y



Project Details

Actionitem Id

Project

Actionitem Created By

Actionitem Assigned To

Actionitem Name

Actionitem Description

Actionitem Status

Milestone Yn: Y

Milestone Date: 08-May-19

Project Details

Actionitem Id

Project

Actionitem Created By

Actionitem Assigned To

Actionitem Name

Actionitem Description

Actionitem Status

Milestone Yn: N

Milestone Date

Milestone Date is disabled when action item is not a milestone, that is Milestone Yn = N

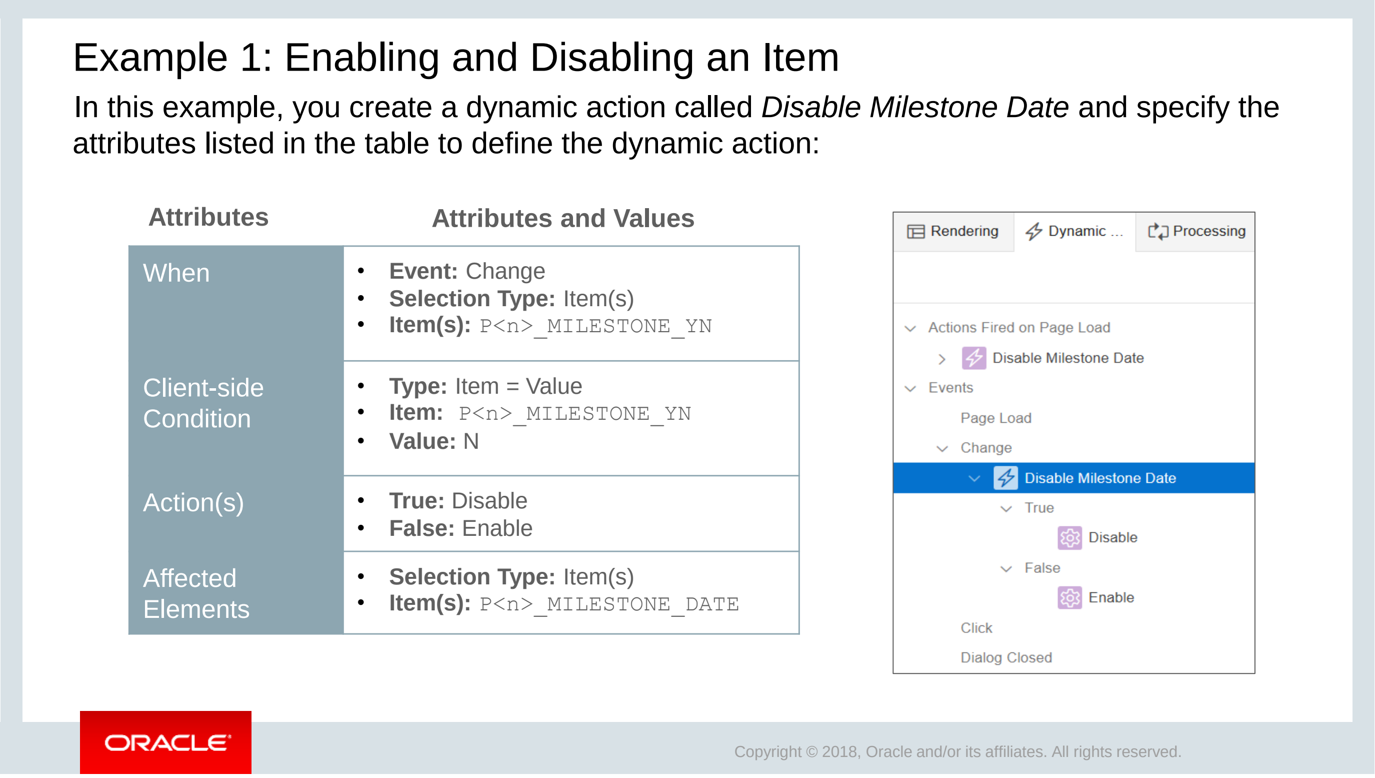
Dynamic Action provides you a way to define client-side behavior declaratively without the need to know JavaScript. You can create a dynamic action from the **Dynamic Actions** tab in Page Designer. From its property editor, you can specify an action that is performed when a defined set of conditions occur. You can also specify the elements that are affected by the action and when and how they are affected.

The process of implementing a dynamic action involves the following:

- Edit or create an item, button, region, DOM object, or jQuery selector on a page. This component is referenced within the dynamic action, which is defined when it fires.
- Create a dynamic action from the application page that invokes the action.
- Run your application to test the dynamic action.

In the PTS application, there are several fields that are dependent on the value in another field. One such instance is the **Milestone Date** field in the *Project Members* form that requires a value only when there is a milestone associated with it. In the example here, you can see that in the screenshot on the left, the value for **Milestone Yn** is `Y` and the **Milestone Date** field is enabled. Whereas in the other screenshot on the right, the value for **Milestone Yn** is `N` and the **Milestone Date** field is disabled. The way in which the logic behind this behavior works is determined by the dynamic action definition.

Many dynamic actions are available in Oracle Application Express. In this course, you examine a few of them. To learn more, review the Oracle Application Express User's Guide. In addition, an OBE tutorial is available in the Oracle Learning Library. This topic is also discussed in detail in the *Advanced APEX Workshop* course.



Example 1: Enabling and Disabling an Item

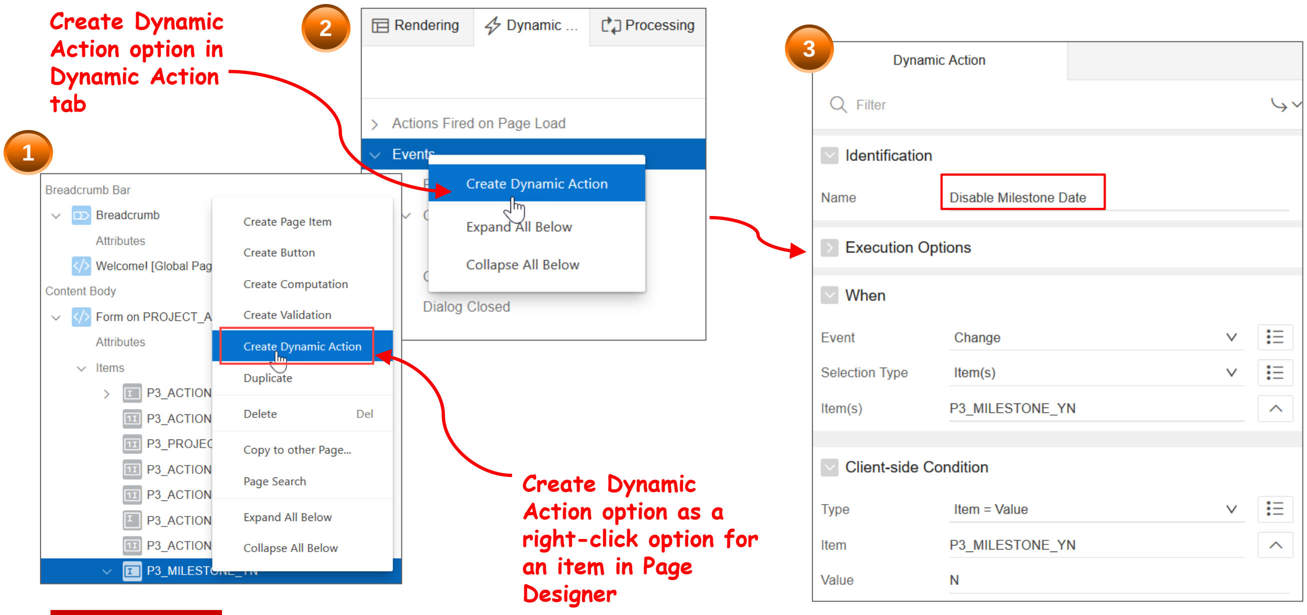
In this example, you create a dynamic action called *Disable Milestone Date* and specify the attributes listed in the table to define the dynamic action:

Attributes	Attributes and Values
When	• **Event:** Change • **Selection Type:** Item(s) • **Item(s):** `P<n>_MILESTONE_YN`
Client-side Condition	• **Type:** Item = Value • **Item:** `P<n>_MILESTONE_YN` • **Value:** N
Action(s)	• **True:** Disable • **False:** Enable
Affected Elements	• **Selection Type:** Item(s) • **Item(s):** `P<n>_MILESTONE_DATE`

ORACLE

The table in the slide lists the values that Steve entered to define the dynamic action so that the Milestone Date field is disabled when the value of `MILESTONE_YN` is equal to N. In addition, you see the dynamic action in the tree view after the *Disable Milestone Date* dynamic action is created.

General Steps to Create a Dynamic Action



Create Dynamic Action option in Dynamic Action tab

Create Dynamic Action option as a right-click option for an item in Page Designer

Name: Disable Milestone Date

Event: Change

Selection Type: Item(s)

Item(s): P3_MILESTONE_YN

Type: Item = Value

Item: P3_MILESTONE_YN

Value: N

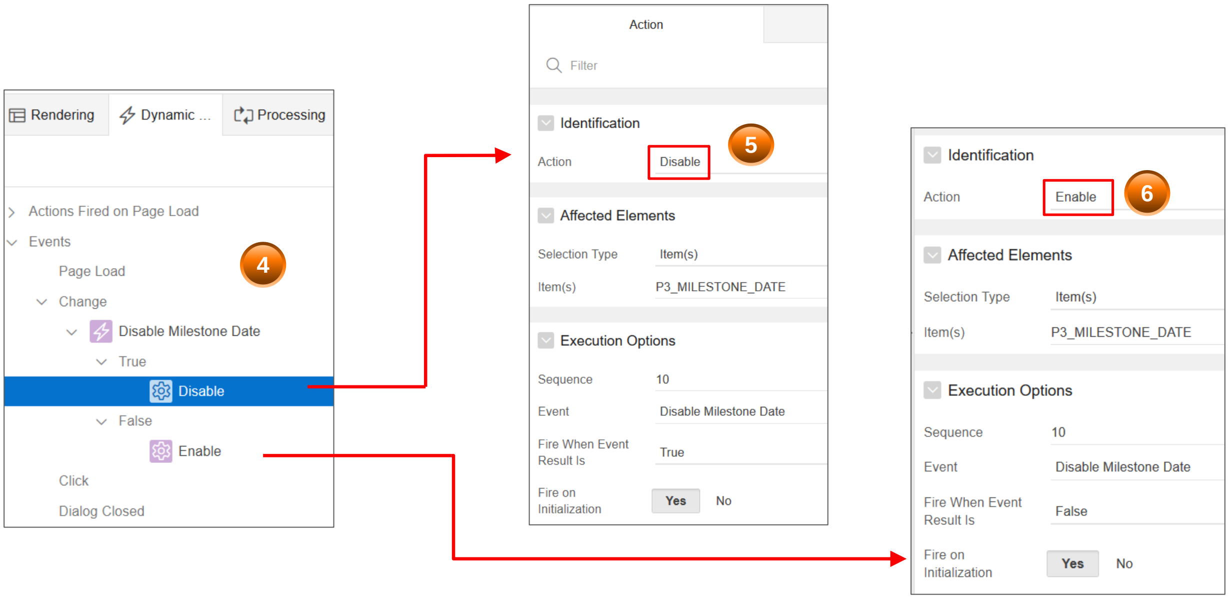
Let us look in detail on how to create a dynamic action. There are two ways to create a dynamic action:

- From a specific item. For example, you can right-click an item **P<n>_MILESTONE_YN** and select **Create Dynamic Action,** as shown in the screenshot.
- From the **Dynamic Actions** tab on Page Designer, as shown in screenshot 2. If you create it from the **Dynamic Actions** tab, you can identify multiple triggering items in the **When** section of the Property Editor, as shown in screenshot 3. The screenshot shows the attributes under **When** that is defined for the *Disable Milestone Date* dynamic action.

Steve wants to enhance his application by creating dynamic actions to perform a few actions automatically to ensure that correct, meaningful data is entered into the PTS application.

In this slide, you can see the *Disable Milestone Date* dynamic action that Steve created. It ensures that the **MILESTONE_DATE** field is disabled when the value for **Milestone** is *N* (No).

General Steps to Create a Dynamic Action



4

- Actions Fired on Page Load
- Events
  - Page Load
  - Change
    - Disable Milestone Date
      - True
        - Disable
      - False
        - Enable
  - Click
  - Dialog Closed

5

Action

Identification

Action	Disable

Affected Elements

Selection Type	Item(s)
Item(s)	P3_MILESTONE_DATE

Execution Options

Sequence	10
Event	Disable Milestone Date
Fire When Event Result Is	True
Fire on Initialization	Yes No

6

Identification

Action	Enable

Affected Elements

Selection Type	Item(s)
Item(s)	P3_MILESTONE_DATE

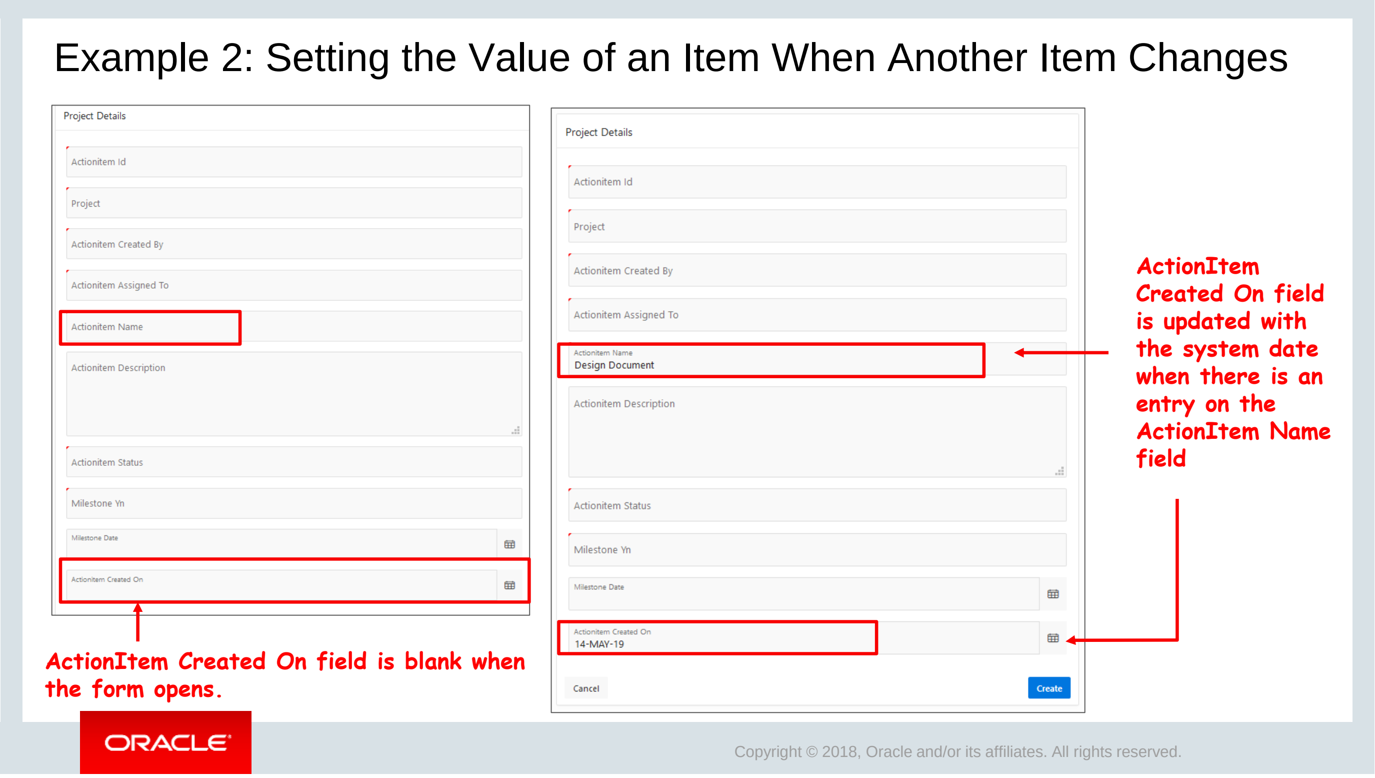
Execution Options

Sequence	10
Event	Disable Milestone Date
Fire When Event Result Is	False
Fire on Initialization	Yes No

Steve also defined the conditions that determine when the **MILESTONE DATE** field is enabled and disabled. As you can see in the screenshots, he defined these conditions under **TRUE** and **FALSE** and calls these actions *Disable* and *Enable* respectively.

For **TRUE** (see screenshot 5), the action is to disable the **MILESTONE DATE** field when **Fire When Event Result is = True.** That is, as per the dynamic action definition in screenshot 2 in the previous slide, when `Item = Value (P3_MILESTONE_YN= N)` in the Client-Side Condition section, then the Event (change) fires. The same logic applies for **FALSE** (see screenshot 6). The logic behind the action for F**ALSE** is to enable the **MILESTONE DATE** field when **Fire When Event Result is = False.**

Example 2: Setting the Value of an Item When Another Item Changes



Actionitem Name: Design Document

Actionitem Created On: 14-MAY-19

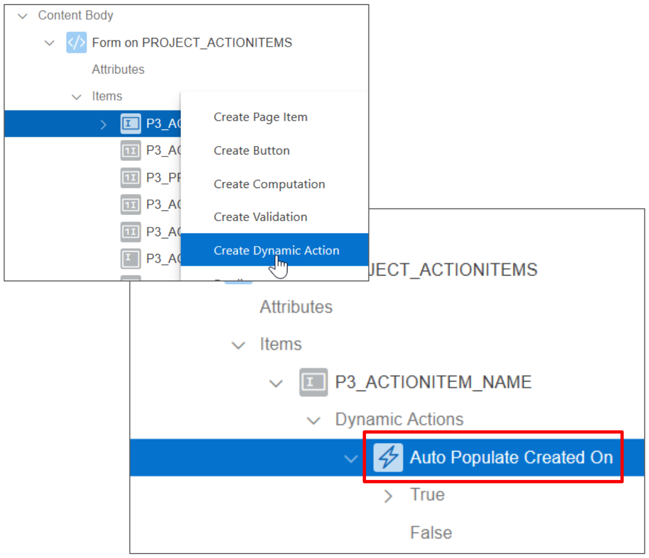
ActionItem Created On field is blank when the form opens.

ActionItem Created On field is updated with the system date when there is an entry on the ActionItem Name field

The example in the slide has a dynamic action that fires when a value is entered in the **ActionItem Name** field. The dynamic action uses a SQL statement to update the **ActionCreated On** field with the current system date when there is an entry in the **ActionItem Name** field.

Example 2: Setting the Value of an Item When Another Item Changes

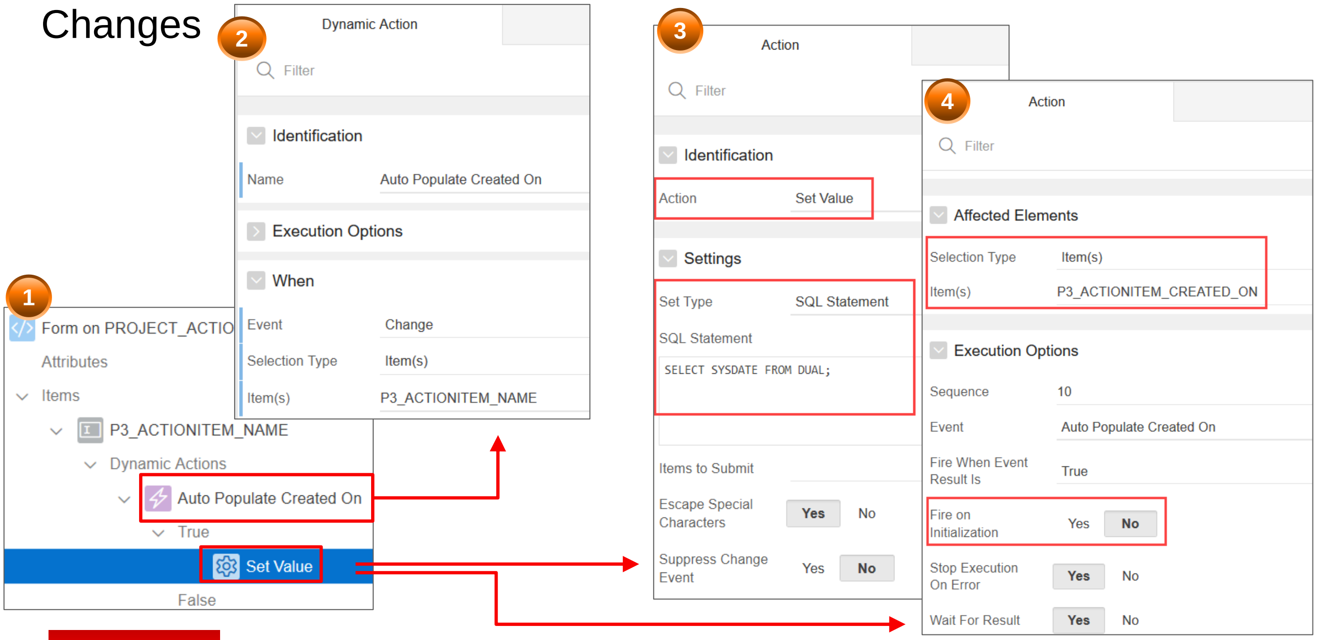
Attributes	Attributes and Values
When	**Event:** Change **Selection Type:** Item(s) **Item(s):** P<n>_ACTIONITEM_NAME Condition: none
Action(s)	**Action:** Set Value
Settings	**Set Type:** SQL Statement **SQL Statement:** `SELECT SYSDATE FROM DUAL;` Page Items to Submit: None
Affected Elements	**Selection Type:** Item(s) **Item(s):** P<n>_ACTIONITEM_CREATED_ON



To create the dynamic action explained in the previous slide, perform the following steps:

1. Open the page that contains the item in Page Designer, right-click the item, and select **Create Dynamic Action.** In this example, dynamic action is created on the item `P3_ACTIONITEM_NAME` in the PTS application.
   The second screenshot on the right shows the *Auto Populate Created On* dynamic action after it is created.

Example 2: Setting the Value of an Item When Another Item Changes



1

Form on PROJECT_ACTIO
Attributes
Items
P3_ACTIONITEM_NAME
Dynamic Actions
Auto Populate Created On
True
Set Value
False

2

Dynamic Action

Identification
Name: Auto Populate Created On

Execution Options

When
Event: Change
Selection Type: Item(s)
Item(s): P3_ACTIONITEM_NAME

3

Action

Identification
Action: Set Value

Settings
Set Type: SQL Statement
SQL Statement:

```
SELECT SYSDATE FROM DUAL;
```

Items to Submit
Escape Special Characters: Yes
Suppress Change Event: No

4

Action

Affected Elements
Selection Type: Item(s)
Item(s): P3_ACTIONITEM_CREATED_ON

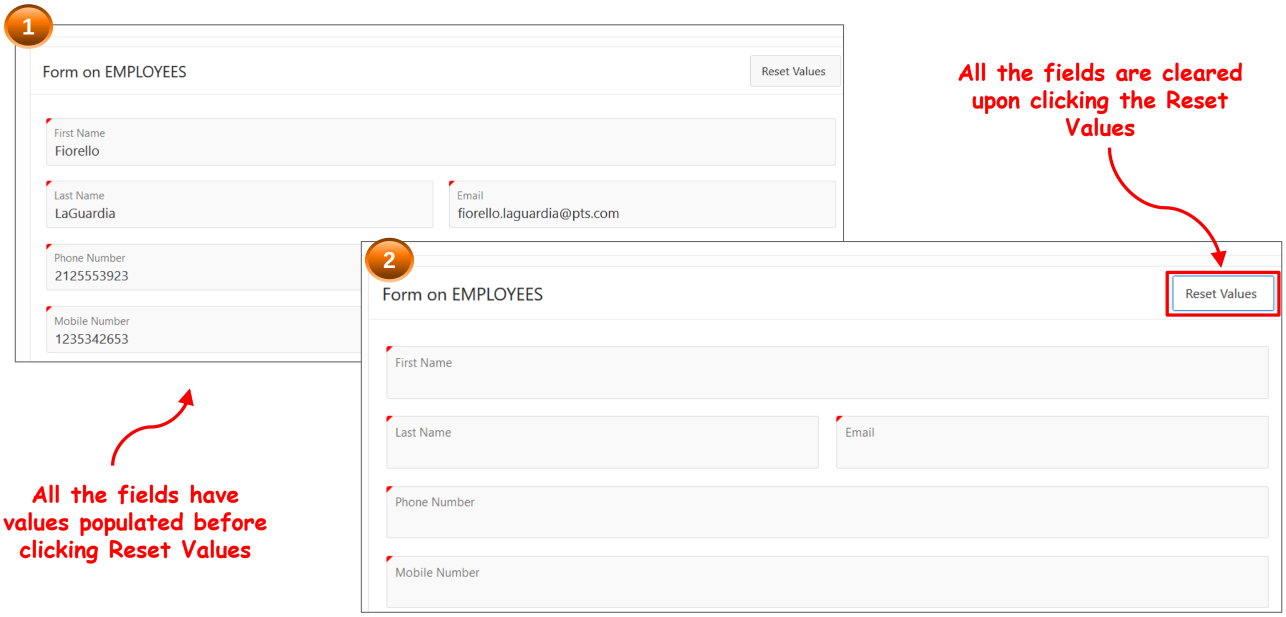
Execution Options
Sequence: 10
Event: Auto Populate Created On
Fire When Event Result Is: True
Fire on Initialization: No
Stop Execution On Error: Yes
Wait For Result: Yes

ORACLE

On the Property Editor of the dynamic action, enter the following values:

2. Enter relevant values such as **Name, Event, Selection Type, Items,** and **Condition** (if any) and value for the condition. In this case, because you created dynamic action from the item directly, you can see that Item is prepopulated.
3. Define the item that is affected by the dynamic action. That is, define what action should be triggered in case the above defined event and condition takes place. To do this, click **True** in the dynamic actions tree under the new dynamic action and update its properties in the Property Editor. You must update the attributes under the **Identification, Settings, Affected Elements**, and **Execution Options** section in the Property Editor, as shown in screenshots 3 and 4 in the slide.
4. **Save** and **Run** the page.

Example 3: Clearing All Items on the Click of a Button



1

Form on EMPLOYEES

First Name: Fiorello

Last Name: LaGuardia

Email: fiorello.laguardia@pts.com

Phone Number: 2125553923

Mobile Number: 1235342653

2

Form on EMPLOYEES

First Name

Last Name

Email

Phone Number

Mobile Number

All the fields are cleared upon clicking the Reset Values

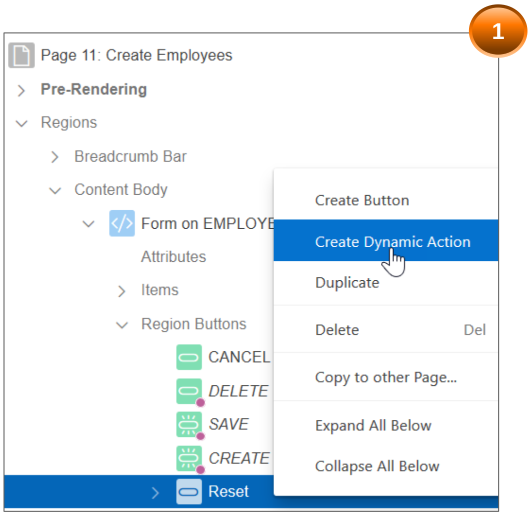
All the fields have values populated before clicking Reset Values

The example in the slide uses a dynamic action *Reset_DynamicAction* that fires whenever you click the **Reset Values** button on the form. As soon as you click **Reset Values,** the dynamic action fires and clears the values from all the fields on the form. In the first screenshot on the left, it shows the form with field values when it opens. The second screenshot on the right shows that all the fields are cleared when the Reset Values button is clicked.

Example 3: Clearing All Items When a Button Is Clicked

To create this dynamic action, specify the following in the wizard:

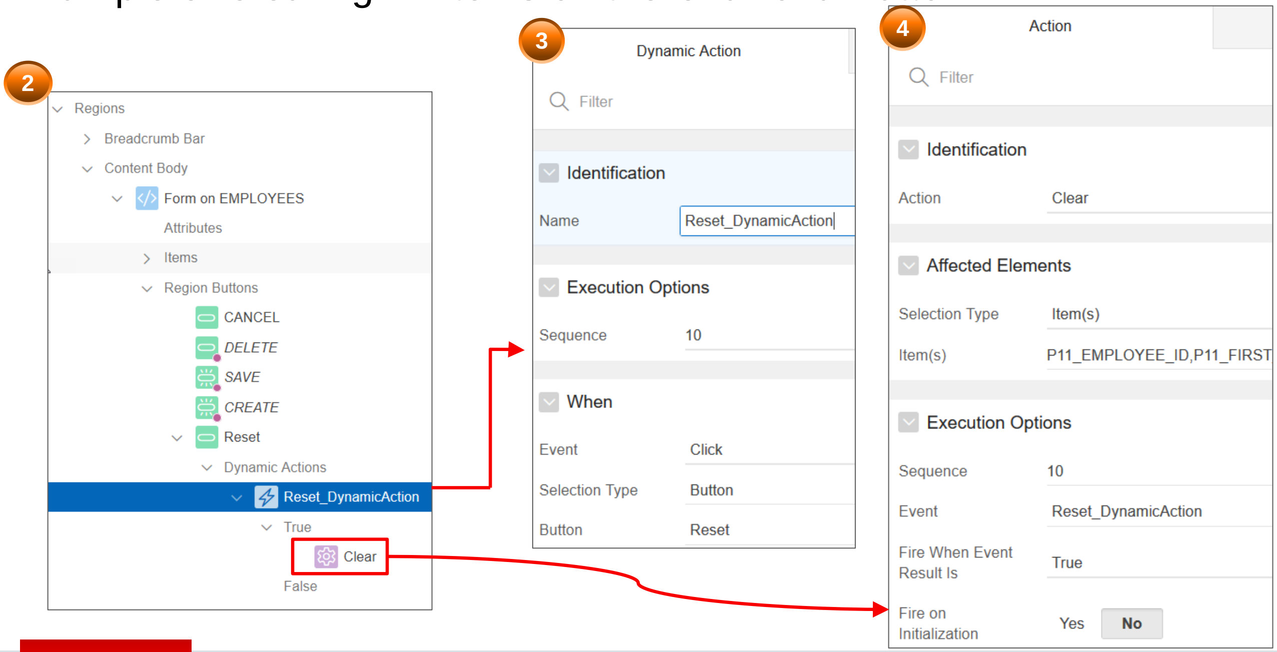
Attributes	Attributes and Values
When	• **Event:** Click • **Selection Type:** Button • **Button:** Reset
Action(s)	• **True:** Clear • **False:** No false action
Affected Elements	• **Selection Type:** Item(s) • **Item(s):** `P11_EMPLOYEE_ID,P11_FIRST_NAME,P11_LAST_NAME,P11_EMAIL,P11_PHONE_NUMBER,P11_MOBILE_NUMBER,P11_ADDRESS,P11_DESIGNATION,P11_SALARY,P11_MANAGER_ID,P11_HIRE_DATE`
Execution Options	• **Fire when Event Result is:** True • **Fire on Initialization:** No



The *Reset_DynamicAction* dynamic action is created on the **Create Employees** form in the PTS application. The dynamic action includes the **Clear** action. To create this type of dynamic action, perform the following steps:

1. Open the page that contains the button in Page Designer, right-click the button, and select **Create Dynamic Action.** In this example, dynamic action is created on the **`Reset`** button (see screenshot 1).
2. Enter the relevant values under **Identification** and **When** sections:
   - **Name:** Reset_DynamicAction
   - **Event:** Click
   - **Selection Type:** Button (Prepopulated)
   - **Button:** Reset (Prepopulated). In this case, because you created the dynamic action from the Reset button directly, you can see that the Selection Type and Button fields are prepopulated with the values Button and Reset, respectively (see the table in the slide).

Example 3: Clearing All Items on the Click of a Button



Dynamic Action

Filter

Identification

Name: Reset_DynamicAction

Execution Options

Sequence: 10

When

Event: Click

Selection Type: Button

Button: Reset

Action

Filter

Identification

Action: Clear

Affected Elements

Selection Type: Item(s)

Item(s): P11_EMPLOYEE_ID,P11_FIRST

Execution Options

Sequence: 10

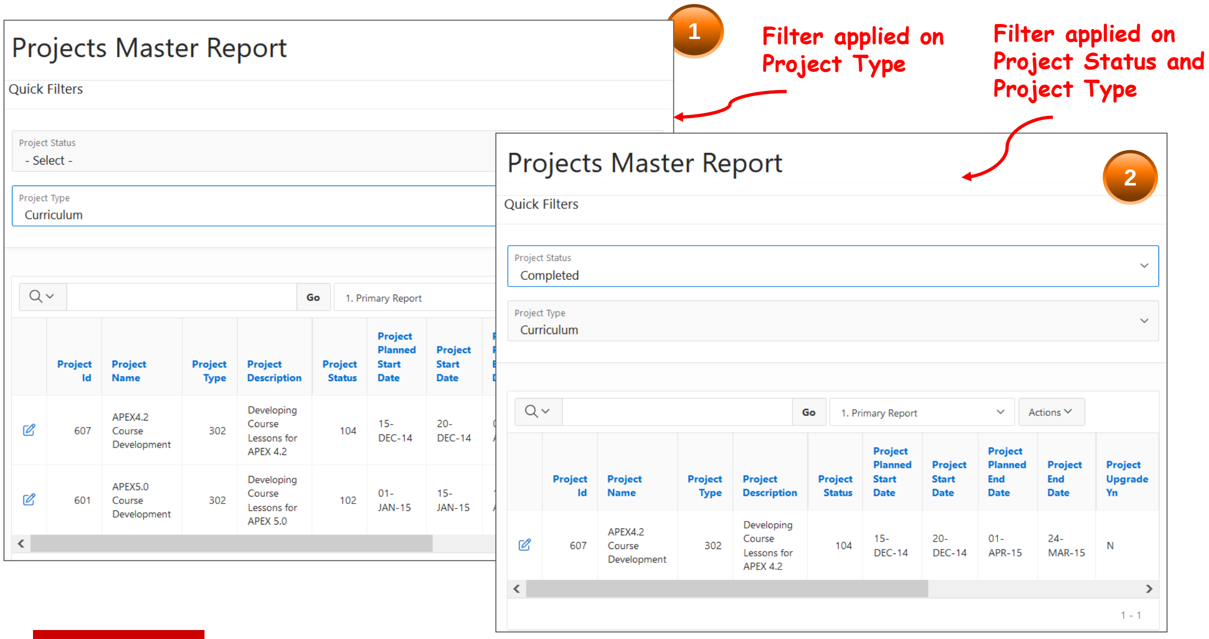
Event: Reset_DynamicAction

Fire When Event Result Is: True

Fire on Initialization: Yes [✓ No]

3. Now, define the items that are affected by the dynamic action. That is, define the action and conditions that would trigger the event (click). To do this, click **True** and then click **Show** under True in the Dynamic Actions tree under the new dynamic action and update its properties in the Property Editor, as shown in screenshot 4.
   - **Action:** Clear (Note: On clicking **Show** under **True,** you set the action name to Clear here, which is seen in screenshot 2)
   - **Selection Type:** Item(s)
   - **Item(s):** `P11_EMPLOYEE_ID, P11_FIRST_NAME, P11_LAST_NAME, P11_EMAIL, P11_PHONE_NUMBER, P11_MOBILE_NUMBER, P11_ADDRESS, P11_DESIGNATION, P11_SALARY, P11_MANAGER_ID, P11_HIRE_DATE.` The values in all these items or fields in the form will be cleared when the dynamic action fires.
   - **Event:** Reset_DynamicAction
   - **Fire When Event Result is:** True
   - **Fire on Initialization:** No. This means that when the form opens, the dynamic action will not fire. The dynamic action will fire only when the condition **Fire When Event Result Is = True** takes place. That is, when the user clicks the Reset button (the event defined in step 2).
     Note: In this example, the FALSE action is not required because the only condition that is applicable is TRUE, which is the click on **Reset.**
4. **Save** and **Run** the page. After the *Form on Employees* page opens with prepopulated values for the selected project, click **Reset Values** located on the top right side of the form. You see that all the items are cleared and does not hold any values after clicking this button.

Example 4: Refreshing the Data in a Report Using Custom Filters


1
Filter applied on Project Type
Filter applied on Project Status and Project Type
2
Quick Filters
Go
1. Primary Report
Actions
1 - 1

Projects Master Report

Quick Filters

Project Status: - Select -

Project Type: Curriculum

Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date
607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14
601	APEX5.0 Course Development	302	Developing Course Lessons for APEX 5.0	102	01-JAN-15	15-JAN-15

Projects Master Report

Quick Filters

Project Status: Completed

Project Type: Curriculum

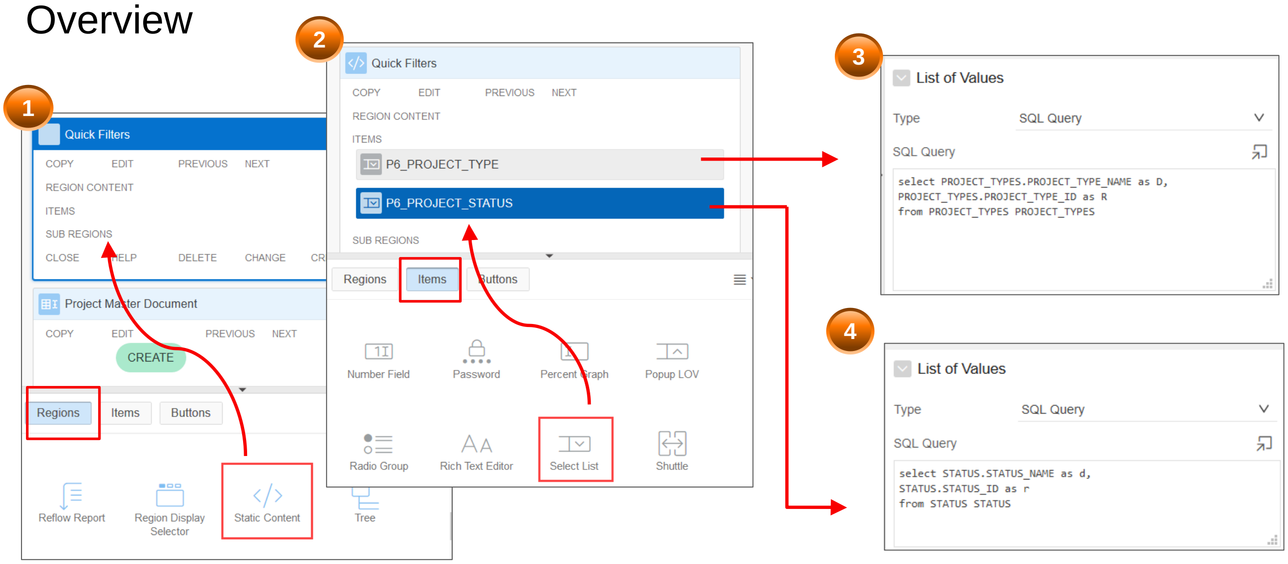
Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date	Project End Date	Project Upgrade Yn
607	APEX4.2 Course Development	302	Developing Course Lessons for APEX 4.2	104	15-DEC-14	20-DEC-14	01-APR-15	24-MAR-15	N

To enhance the usability of the interactive report in the PTS application, Steve decides to create an AJAX-based dynamic action on the *Project Master Report,* which is an interactive report in the application. First, he adds a new region *Quick Filters* with Select Lists for Project Type and Project Status. Next, he adds a dynamic action for report refresh when a value is selected from the quick filters.

This is the last example on dynamic action that demonstrates how custom filters can be added to reports very easily with dynamic actions. Dynamic actions can handle AJAX-based filtering of report data. The approach is slightly different depending on whether your report is a classic or an interactive report.

In this example, when either a **Project Type** or **Project Status** is selected in the report, the report is refreshed by AJAX to show the refreshed values. To implement this, a dynamic action is defined on the report that refreshes the report after a value is selected for **Project Type** or **Project Status.** To accomplish this, it uses the **Refresh** action.

Example 4: Refreshing the Data in a Report Using Custom Filters: Overview



List of Values

Type: SQL Query

SQL Query:

```
select PROJECT_TYPES.PROJECT_TYPE_NAME as D,
PROJECT_TYPES.PROJECT_TYPE_ID as R
from PROJECT_TYPES PROJECT_TYPES
```

List of Values

Type: SQL Query

SQL Query:

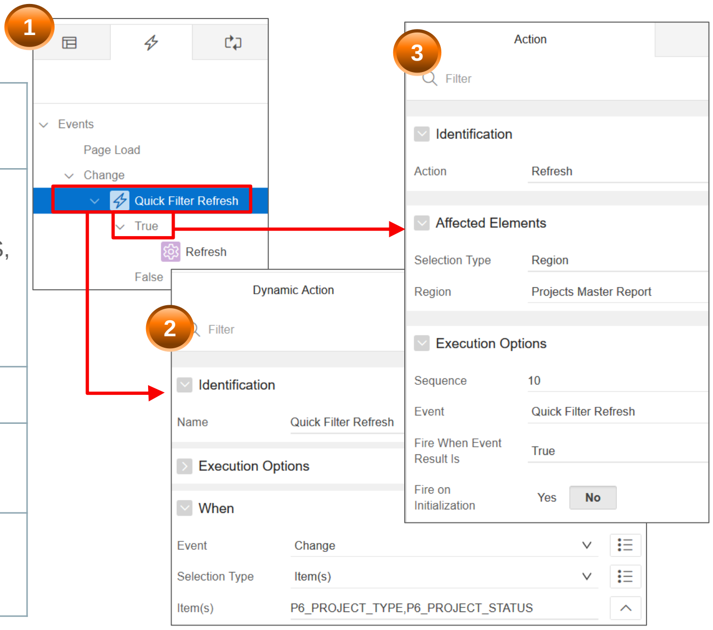
```
select STATUS.STATUS_NAME as d,
STATUS.STATUS_ID as r
from STATUS STATUS
```

First, let’s create the *Quick Filter* region on the *Project Master Report* (created as part of Lesson 5) and then add the AJAX-based dynamic action on the items. To create this dynamic action, perform the following steps:

1. Open the *Project Master Report* in Page Designer. Drag the **Static Content** from **Regions** gallery and drop it in Content Body above Projects Master Report region in grid layout, as shown in screenshot 1.
2. A new region is created, and its properties are shown in Property Editor. In the Property Editor, enter `Quick Filters` for its **Name.**
3. Drag two **Select List** items from **Items** gallery and drop it in **Quick Filters** region in grid layout, as shown in screenshot 2.
4. For the first Select List, enter the following (see screenshot 3):
   - **Name:** `P6_PROJECT_TYPE`
   - **Type** (under List of Values): **SQL Query**
   - **SQL Query:**
     ```
     select PROJECT_TYPES.PROJECT_TYPE_NAME as D,
             PROJECT_TYPES.PROJECT_TYPE_ID as R
             from PROJECT_TYPES PROJECT_TYPES
     ```
5. For the second Select List, enter the following (see screenshot 3) and click **Save**:
   - **Name:** P6_PROJECT_STATUS
   - **Type** (under List of Values): **SQL Query**
   - **SQL Query:**
     ```
     select STATUS.STATUS_NAME as d,
             STATUS.STATUS_ID as r
             from STATUS STATUS
     ```

Example 4: Refreshing the Data in a Report Using Custom Filters

Attributes	Attributes and Values
Identification	• **Name:** Quick Filter Refresh
When	• **Event:** Change • **Selection Type:** Item(s) • **Item(s):** P<n>_PROJECT_STATUS, P<n>_PROJECT_TYPE
Action(s)	• **True:** Refresh
Affected Elements	• **Selection Type:** Region • **Region:** <report name>
Execution Options	• **Event:** Quick Filter Refresh • **Fire When Event Result Is:** True • **Fire on Initialization:** No



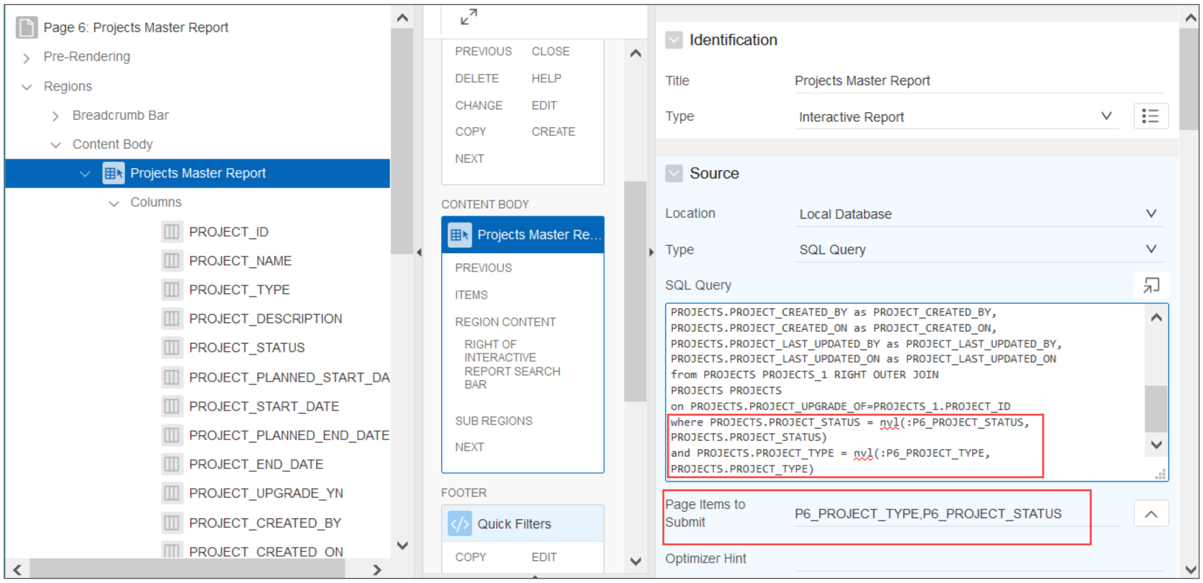
Identification
Name: Quick Filter Refresh
Execution Options
When
Event: Change
Selection Type: Item(s)
Item(s): P6_PROJECT_TYPE,P6_PROJECT_STATUS

Identification
Action: Refresh
Affected Elements
Selection Type: Region
Region: Projects Master Report
Execution Options
Sequence: 10
Event: Quick Filter Refresh
Fire When Event Result Is: True
Fire on Initialization: Yes [✓ No]

Next, create the dynamic actions on the Select List items `P6_PROJECT_STATUS` and `P6_PROJECT_STATUS`.

1. Go to the **Dynamic Actions** tab, right-click **Events,** and select **Create Dynamic Action.**
2. For the dynamic action, update the following values in its Property Editor, as shown in in screenshot 2.
   a. **Name:** Quick Filter Refresh
   b. **Event:** Change
   c. **Selection Type:** Item(s)
   d. **Item(s):** Select `p<n>_PROJECT_STATUS, P<n>_PROJECT_TYPE` for Item(s). These selections define that the dynamic action will fire whenever the value in the Status or Project Type list changes.
3. For **TRUE** action, update the following in the Property Editor, as shown in screenshot 3.
   a. **Action:** Refresh. The Refresh action currently supports interactive report regions, classic reports, and all item types with cascading LOV support. It also supports item or region plug-ins, depending on whether the plug-in author has coded the plug-in accordingly.
   b. **Affected Elements:** For **Selection Type** select Region, and for **Region** select Project Master Report .
   c. **Execution Options:** For **Event,** select **Quick Filter Refresh.**
   d. **Fire When Event Result is:** True
   e. **Fire on Initialization:** No.
4. Click **Save** and **Run.** After the report loads, select a value for **Project Status** and observe the report region is refreshed.

Example 4: Refreshing the Data in a Report Using Custom Filters



Title: Projects Master Report

Type: Interactive Report

Location: Local Database

Type: SQL Query

SQL Query:

```
PROJECTS.PROJECT_CREATED_BY as PROJECT_CREATED_BY,
PROJECTS.PROJECT_CREATED_ON as PROJECT_CREATED_ON,
PROJECTS.PROJECT_LAST_UPDATED_BY as PROJECT_LAST_UPDATED_BY,
PROJECTS.PROJECT_LAST_UPDATED_ON as PROJECT_LAST_UPDATED_ON
from PROJECTS PROJECTS_1 RIGHT OUTER JOIN
PROJECTS PROJECTS
on PROJECTS.PROJECT_UPGRADE_OF=PROJECTS_1.PROJECT_ID
where PROJECTS.PROJECT_STATUS = nvl(:P6_PROJECT_STATUS,
PROJECTS.PROJECT_STATUS)
and PROJECTS.PROJECT_TYPE = nvl(:P6_PROJECT_TYPE,
PROJECTS.PROJECT_TYPE)
```

Page Items to Submit: P6_PROJECT_TYPE,P6_PROJECT_STATUS

Optimizer Hint

Even though the dynamic action fired and the report is refreshed, it is not being scoped by the filter selection. This is because the values for the filter page items are not being saved to session state and are, therefore, not set when the SQL is executed.

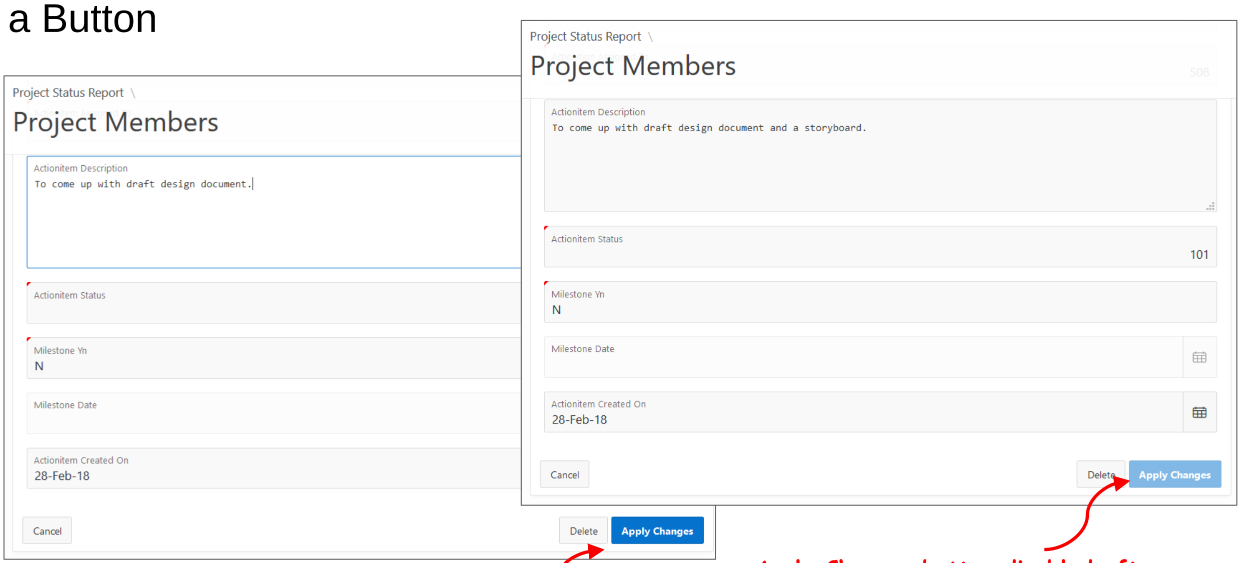
Perform the following steps to define the interactive report region to save the values of these item in session state after the region is refreshed:

1. Select report region under page rendering and select `P<n>_PROJECT_STATUS, P<n>_PROJECT_TYPE` for **Page Items to Submit.**
2. Modify the SQL query under Source to mention `PROJECT_STATUS and PROJECT_TYPE` in the where clause, as shown in the screenshot.

```
    where PROJECTS.PROJECT_STATUS = nvl(:P6_PROJECT_STATUS,
    PROJECTS.PROJECT_STATUS)
    and PROJECTS.PROJECT_TYPE = nvl(:P6_PROJECT_TYPE,
PROJECTS.PROJECT_TYPE)
```

Now if you run the page, you see that the filters are fully functional. Select different **Project Status IDs** and **Project Types** and see the refreshed report, which now shows only those projects that come under the scope of your selections.

Example: Disabling a Button and Submitting the Page When Clicking a Button



Project Status Report \

Project Members

Actionitem Description: To come up with draft design document.

Actionitem Status:

Milestone Yn: N

Milestone Date:

Actionitem Created On: 28-Feb-18

Project Status Report \

Project Members

Actionitem Description: To come up with draft design document and a storyboard.

Actionitem Status: 101

Milestone Yn: N

Milestone Date:

Actionitem Created On: 28-Feb-18

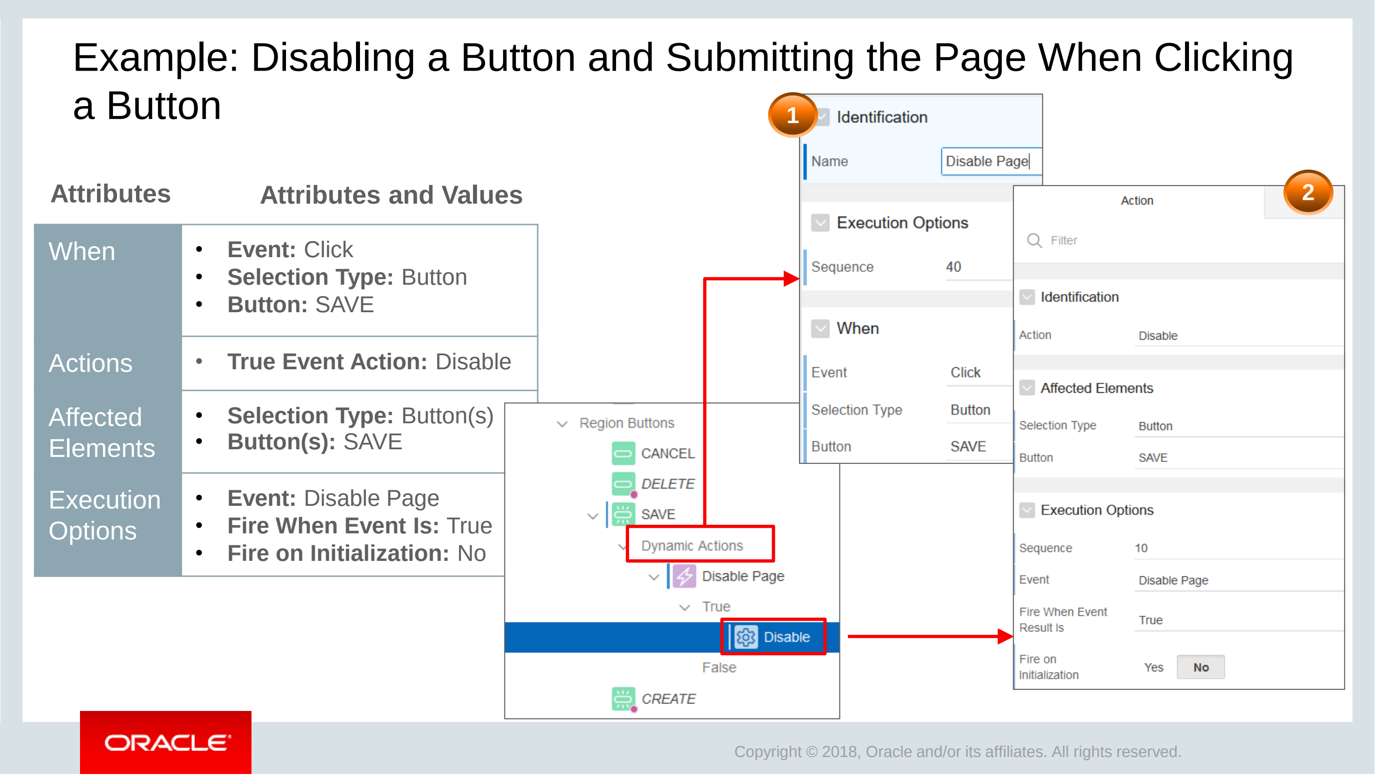
Apply Changes button before clicking it

Apply Changes button disabled after clicking it and page submission

Steve considers a similar situation when a user submits a page by clicking **Apply Changes.** It takes a few seconds for the page processing to complete and the resultant page to be displayed. Within these few seconds, it is possible for the user to click the button again or make more changes to it. This can lead to data integrity issues and also increase the response time if the button is clicked again and again. To avoid this situation, Steve creates a dynamic action to be fired when a button is clicked and which will disable the button clicked so that the user cannot click it again before the page processing is complete. In this example, it is the **Apply Changes** button, as you can see in the screenshot.

The example in the slide shows a form for update. A dynamic action is defined to fire when the **Apply Changes** button is clicked. When the user clicks the button, the button is disabled. After the page processing is complete, the resultant page is displayed.

You can also create a dynamic action to be fired when the user clicks a button, for instance **Save, Apply Changes**, or **Create.**



Example: Disabling a Button and Submitting the Page When Clicking a Button

Attributes	Attributes and Values
When	**Event:** Click **Selection Type:** Button **Button:** SAVE
Actions	**True Event Action:** Disable
Affected Elements	**Selection Type:** Button(s) **Button(s):** SAVE
Execution Options	**Event:** Disable Page **Fire When Event Is:** True **Fire on Initialization:** No

1

Identification

Name: Disable Page

Execution Options

Sequence: 40

When

Event: Click

Selection Type: Button

Button: SAVE

2

Action

Identification

Action: Disable

Affected Elements

Selection Type: Button

Button: SAVE

Execution Options

Sequence: 10

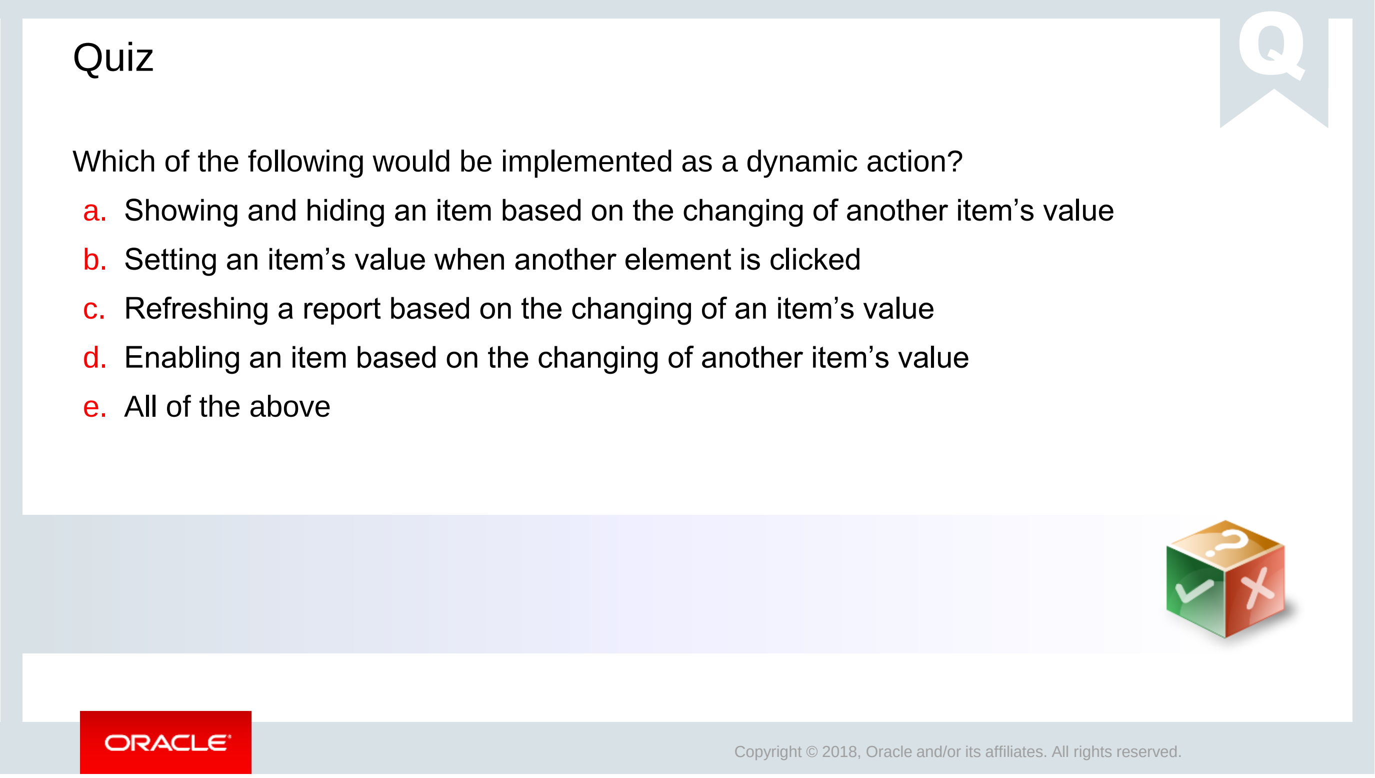
Event: Disable Page

Fire When Event Result Is: True

Fire on Initialization: Yes **No**

To create a dynamic action on a button, perform the following steps:

1. Navigate to the page that contains the Save button. In the tree view under Page Designer, right-click the **Save** button and select **Create Dynamic Action.**
2. Update the properties of dynamic action in its Property Editor (see screenshot 1):
   - **Name:** Enter `Disable Page`
   - **Event:** Click (prepopulated)
   - **Selection Type:** Button (prepopulated)
   - **Button:** SAVE (prepopulated)
3. Define the properties for the TRUE event under Identification and Affected Elements (see screenshot 2):
   - **Action:** Select Disable
   - **Selection Type:** Button
   - **Button:** SAVE
4. Define the Execution Options (see screenshot 2):
   - **Sequence:** 10 (prepopulated)
   - **Event:** Disable Page
   - **Fire when Event Result is:** True
   - **Fire On Initialization:** No
5. Click **Save** and **Run.** After the page loads, make some edits to the form and click **Apply Changes.** You will observe that the **Apply Changes** button gets disabled.

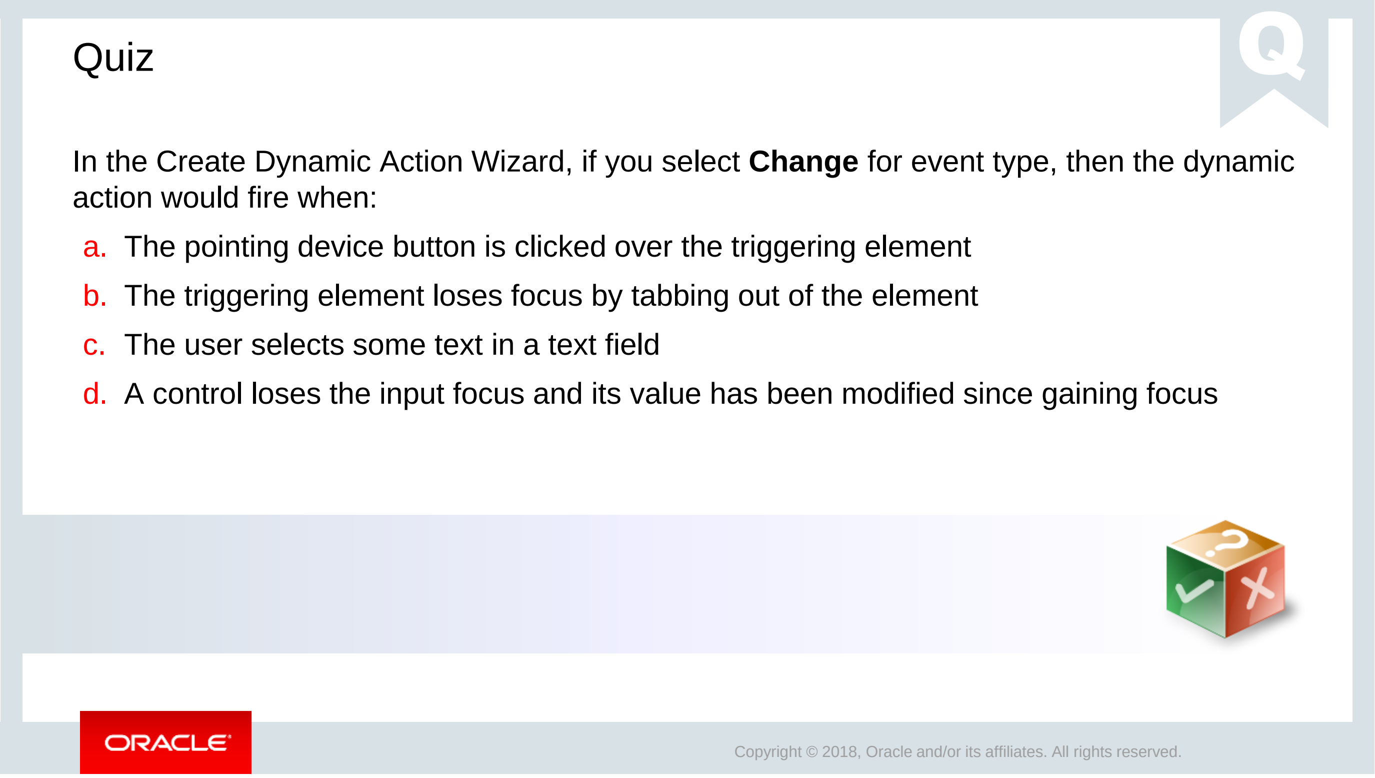


Quiz

Which of the following would be implemented as a dynamic action?

a. Showing and hiding an item based on the changing of another item’s value
b. Setting an item’s value when another element is clicked
c. Refreshing a report based on the changing of an item’s value
d. Enabling an item based on the changing of another item’s value
e. All of the above

Answer: e



Quiz

In the Create Dynamic Action Wizard, if you select **Change** for event type, then the dynamic action would fire when:

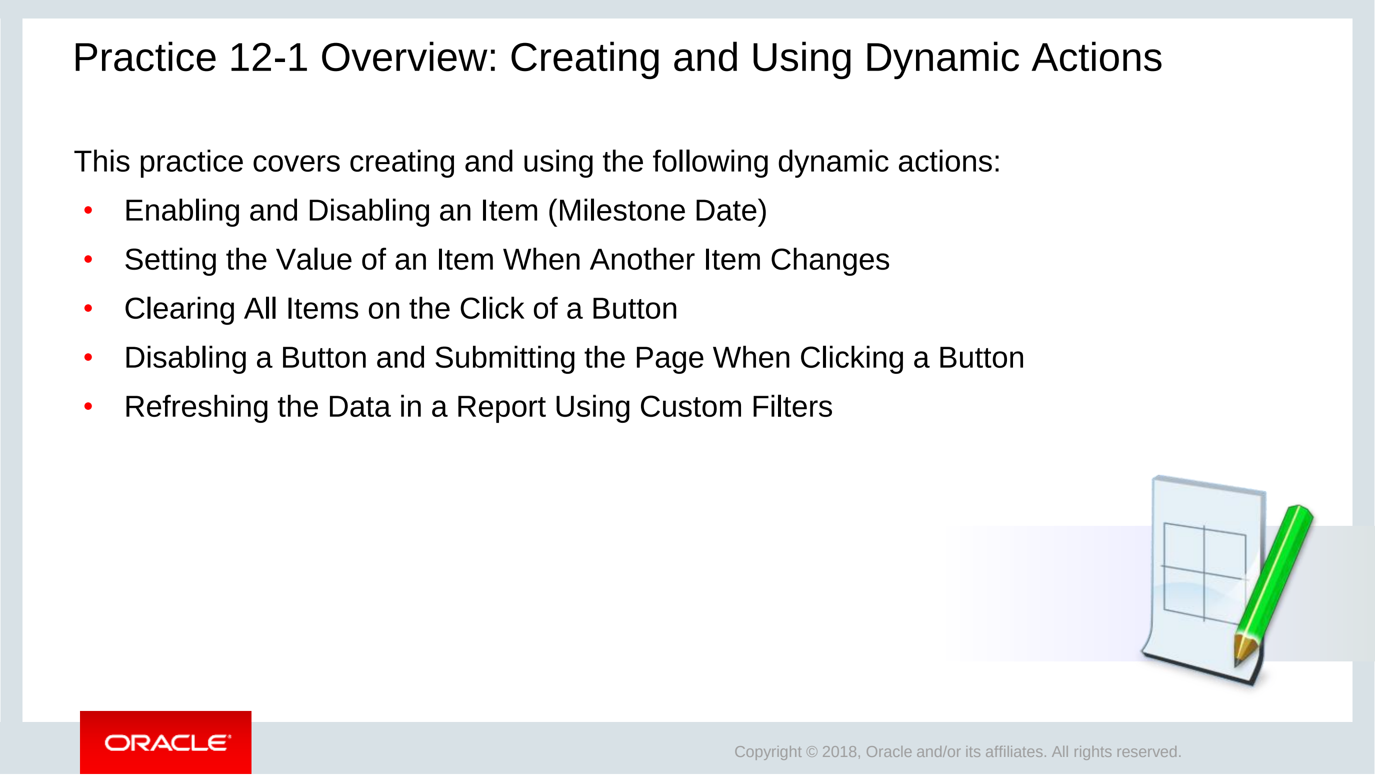
a. The pointing device button is clicked over the triggering element

b. The triggering element loses focus by tabbing out of the element

c. The user selects some text in a text field

d. A control loses the input focus and its value has been modified since gaining focus

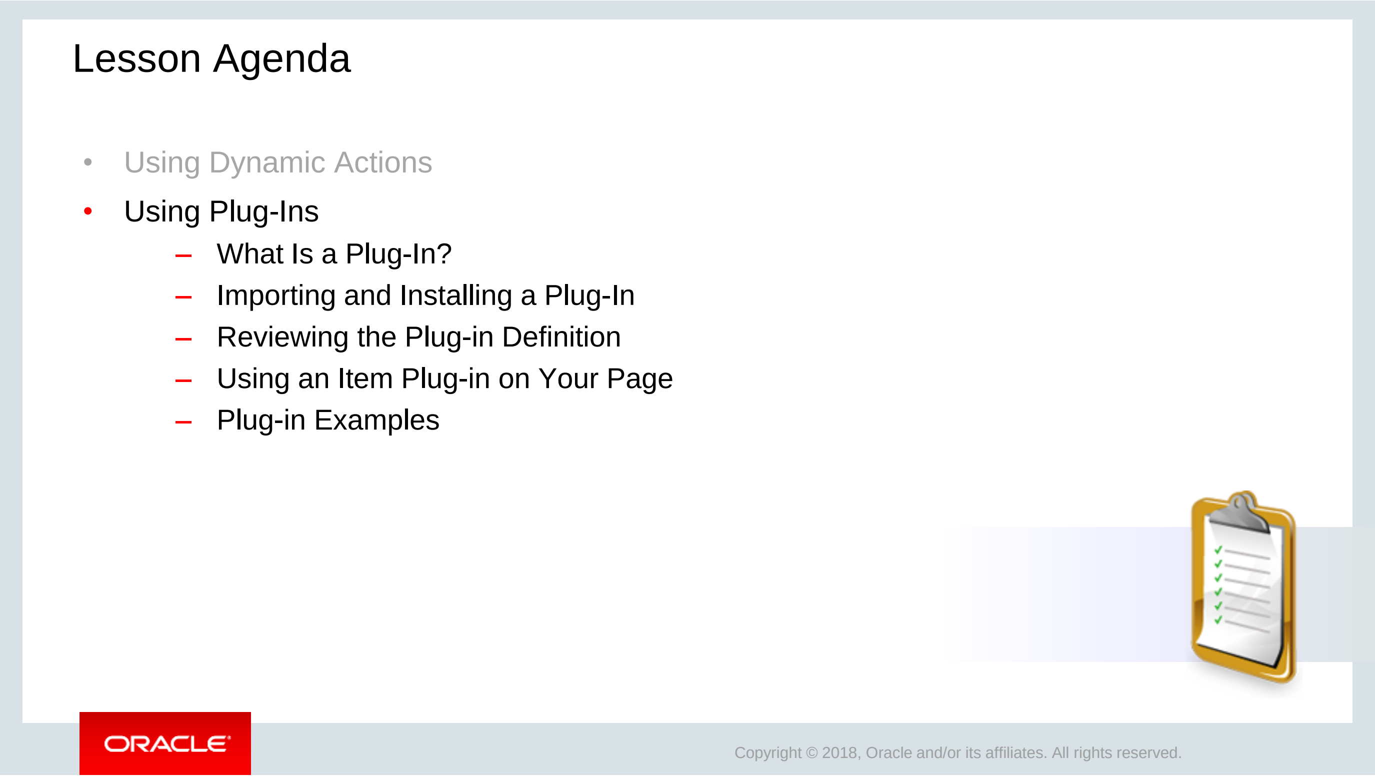
Answer: d



Practice 12-1 Overview: Creating and Using Dynamic Actions

This practice covers creating and using the following dynamic actions:

- Enabling and Disabling an Item (Milestone Date)
- Setting the Value of an Item When Another Item Changes
- Clearing All Items on the Click of a Button
- Disabling a Button and Submitting the Page When Clicking a Button
- Refreshing the Data in a Report Using Custom Filters

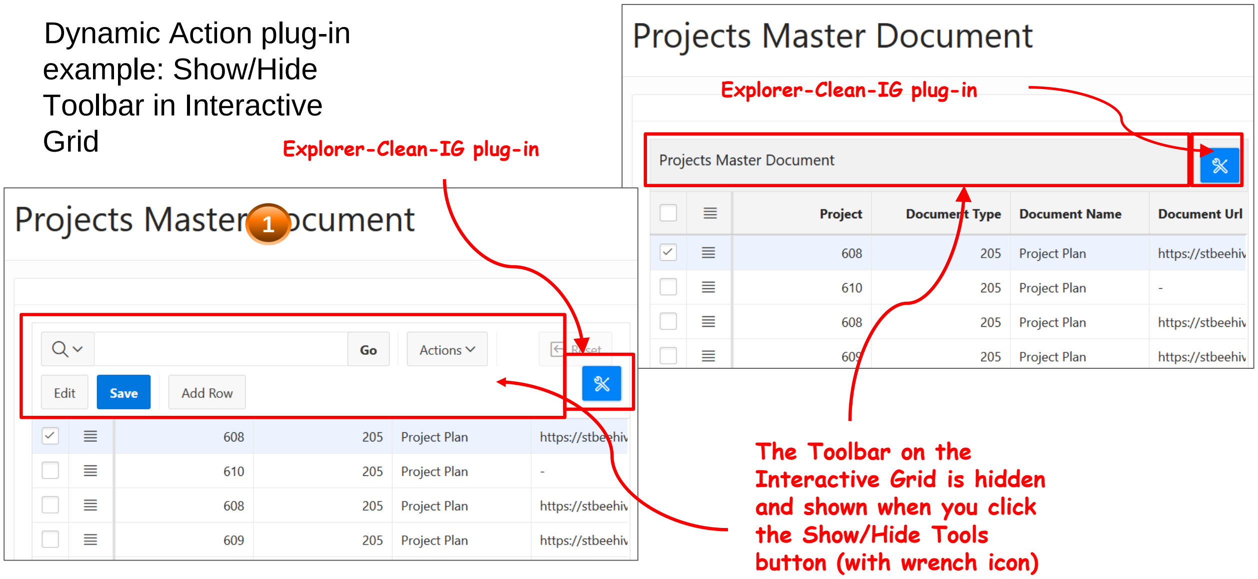


Lesson Agenda

- Using Dynamic Actions
- Using Plug-Ins
  - What Is a Plug-In?
  - Importing and Installing a Plug-In
  - Reviewing the Plug-in Definition
  - Using an Item Plug-in on Your Page
  - Plug-in Examples

What Is a Plug-In?

Dynamic Action plug-in example: Show/Hide Toolbar in Interactive Grid



Explorer-Clean-IG plug-in

Explorer-Clean-IG plug-in

□		Project	Document Type	Document Name	Document Url
☑	≡	608	205	Project Plan	https://stbeehiv
□	≡	610	205	Project Plan	-
□	≡	608	205	Project Plan	https://stbeehiv
□	≡	609	205	Project Plan	https://stbeehiv

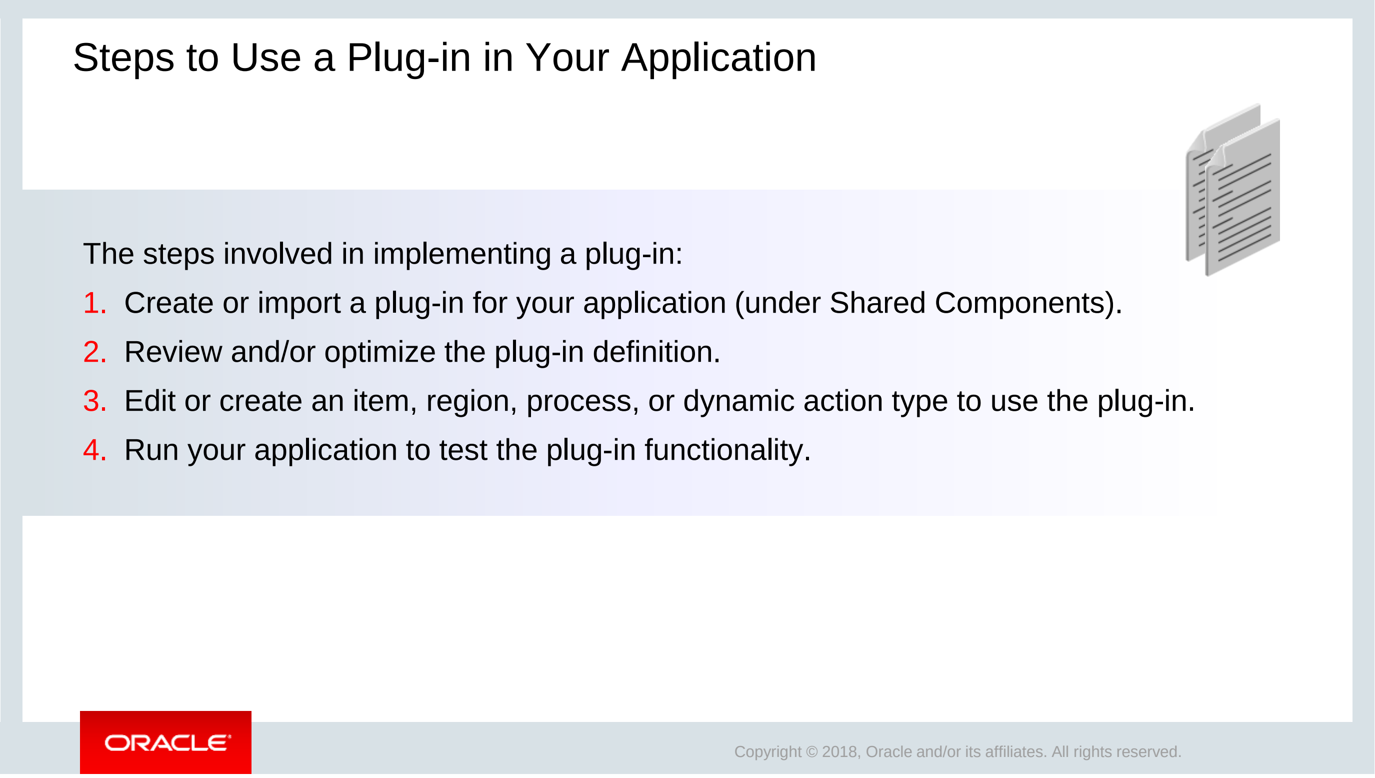
The Toolbar on the Interactive Grid is hidden and shown when you click the Show/Hide Tools button (with wrench icon)

In the screenshots, you can see the **Explorer-Clean-IG** plug-in, depicted by the button with the wrench icon. The plug-in shows and hides the interactive grid toolbar. In the first screenshot, the toolbar is hidden. Clicking the wrench icon shows the interactive grid toolbar, as seen in screenshot 2. If you want to provide your user with different experiences while viewing the interactive grid and while editing it, then this plug-in is a good option as it gives the user the choice to show and hide the toolbar.

Plug-ins enable developers to declaratively extend the built-in types available with Application Express and share and reuse them.

Oracle Application Express supports a set group of item, region, dynamic action, and process types. Plug-ins offer a means of augmenting these built-in types by declaratively creating and using new types in your application. Because plug-ins are designed for reuse, developers can export them from and import them into other applications in the same or other workspaces and also share them with the Application Express plug-in community by using the Plug-in Repository.

There are a number of plug-ins available in the Plug-in Repository (accessed from the plug-in window). To find out more about plug-ins, see the *Application Express User's Guide*.



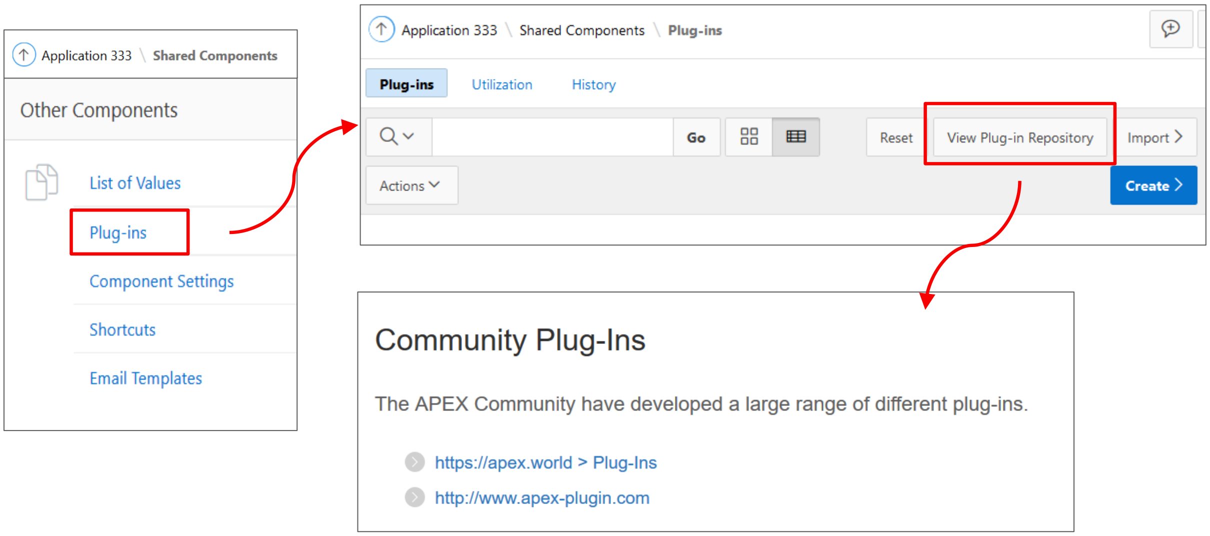
Steps to Use a Plug-in in Your Application

The steps involved in implementing a plug-in:

1. Create or import a plug-in for your application (under Shared Components).
2. Review and/or optimize the plug-in definition.
3. Edit or create an item, region, process, or dynamic action type to use the plug-in.
4. Run your application to test the plug-in functionality.

The slide shows the steps involved in implementing a plug-in in your application. In this lesson, you import plug-ins from the Plug-in Repository, review the plug-in definition, and make some changes to optimize the use of the plug-ins. You will create appropriate objects on your page that will use the plug-ins and run the page to view the results.

Accessing the Plug-in Repository



Community Plug-Ins

The APEX Community have developed a large range of different plug-ins.

- https://apex.world > Plug-Ins
- http://www.apex-plugin.com

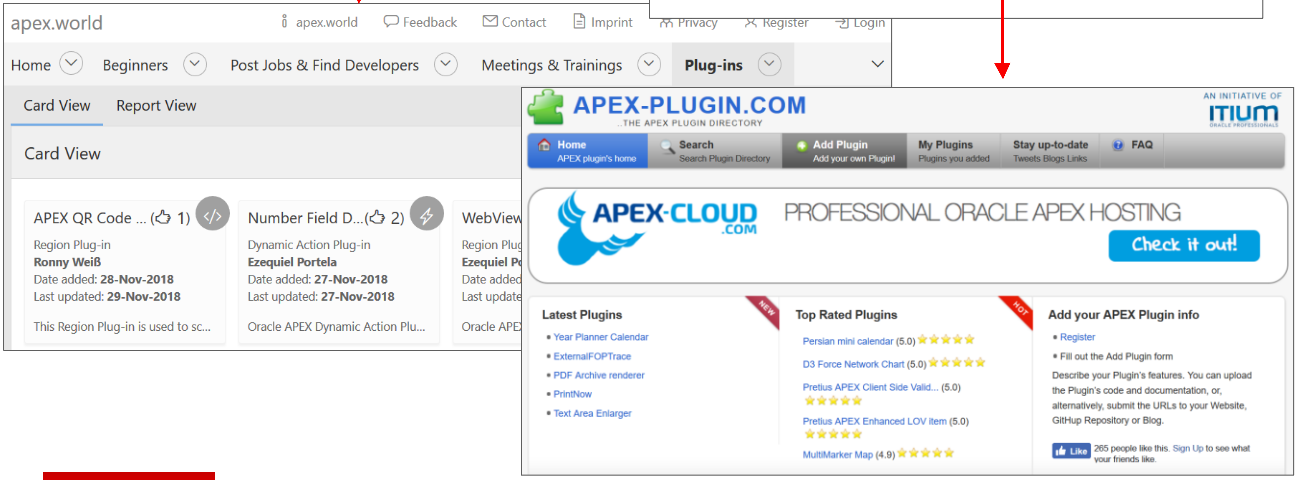
The Plug-in Repository provides a series of available plug-ins developed by Oracle that can be used by customers to perform various tasks. This repository continues to be updated with additional plug-ins for use by the Oracle Application Express user community.

Accessing the Plug-in Repository

Community Plug-Ins

The APEX Community have developed a large range of different plug-ins.

- https://apex.world > Plug-Ins
- http://www.apex-plugin.com



Region Plug-in
Ronny Weiß
Date added: **28-Nov-2018**
Last updated: **29-Nov-2018**

This Region Plug-in is used to sc...

Dynamic Action Plug-in
Ezequiel Portela
Date added: **27-Nov-2018**
Last updated: **27-Nov-2018**

Oracle APEX Dynamic Action Plu...

Latest Plugins

- Year Planner Calendar
- ExternalFOPTrace
- PDF Archive renderer
- PrintNow
- Text Area Enlarger

Top Rated Plugins

Persian mini calendar (5.0) ★★★★★

D3 Force Network Chart (5.0) ★★★★★

Pretius APEX Client Side Valid... (5.0) ★★★★★

Pretius APEX Enhanced LOV item (5.0) ★★★★★

MultiMarker Map (4.9) ★★★★★

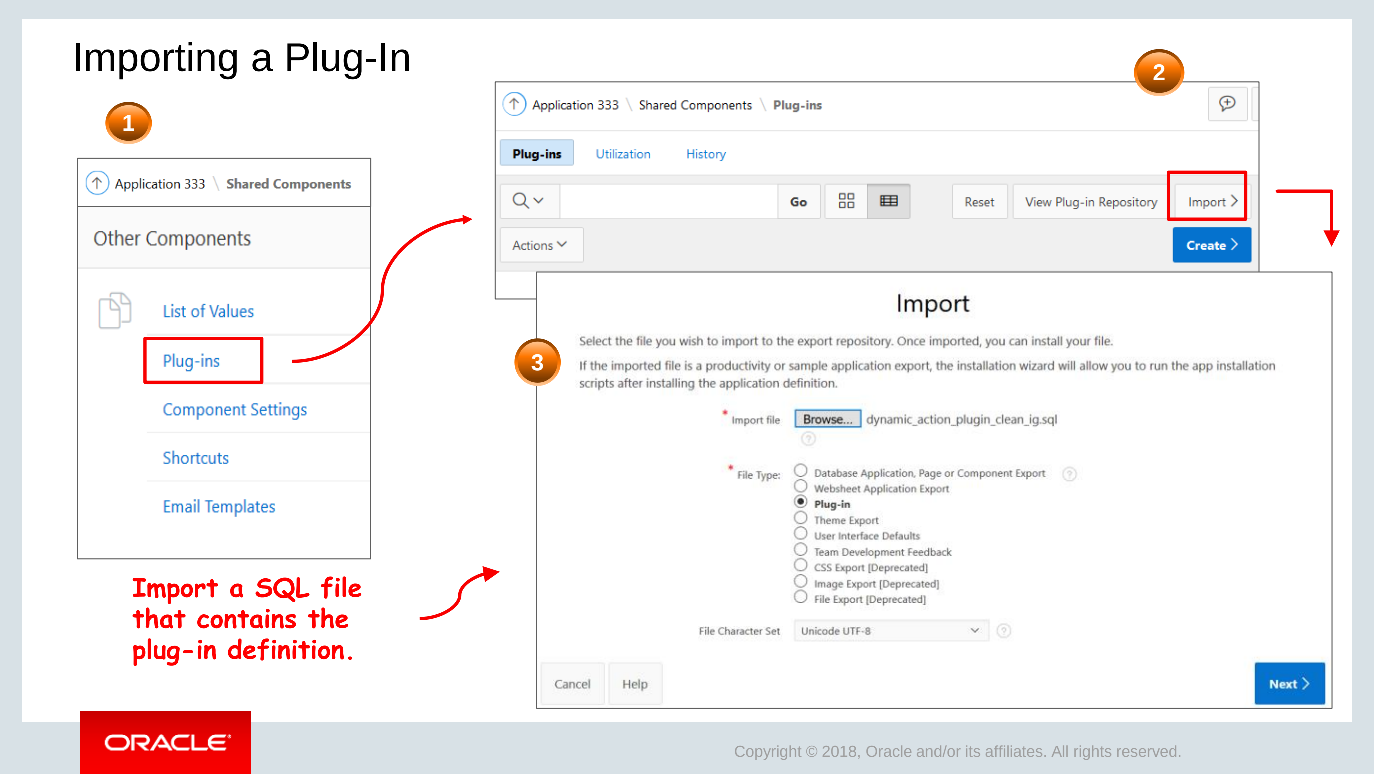
Add your APEX Plugin info

- Register
- Fill out the Add Plugin form

Describe your Plugin's features. You can upload the Plugin's code and documentation, or, alternatively, submit the URLs to your Website, GitHup Repository or Blog.

265 people like this. Sign Up to see what your friends like.

This screenshot shows the Plug-in Repository in apex.world and apex-plugin.com.



Importing a Plug-In

Import

Select the file you wish to import to the export repository. Once imported, you can install your file.

If the imported file is a productivity or sample application export, the installation wizard will allow you to run the app installation scripts after installing the application definition.

* Import file: Browse... dynamic_action_plugin_clean_ig.sql

* File Type:
○ Database Application, Page or Component Export
○ Websheet Application Export
● Plug-in
○ Theme Export
○ User Interface Defaults
○ Team Development Feedback
○ CSS Export [Deprecated]
○ Image Export [Deprecated]
○ File Export [Deprecated]

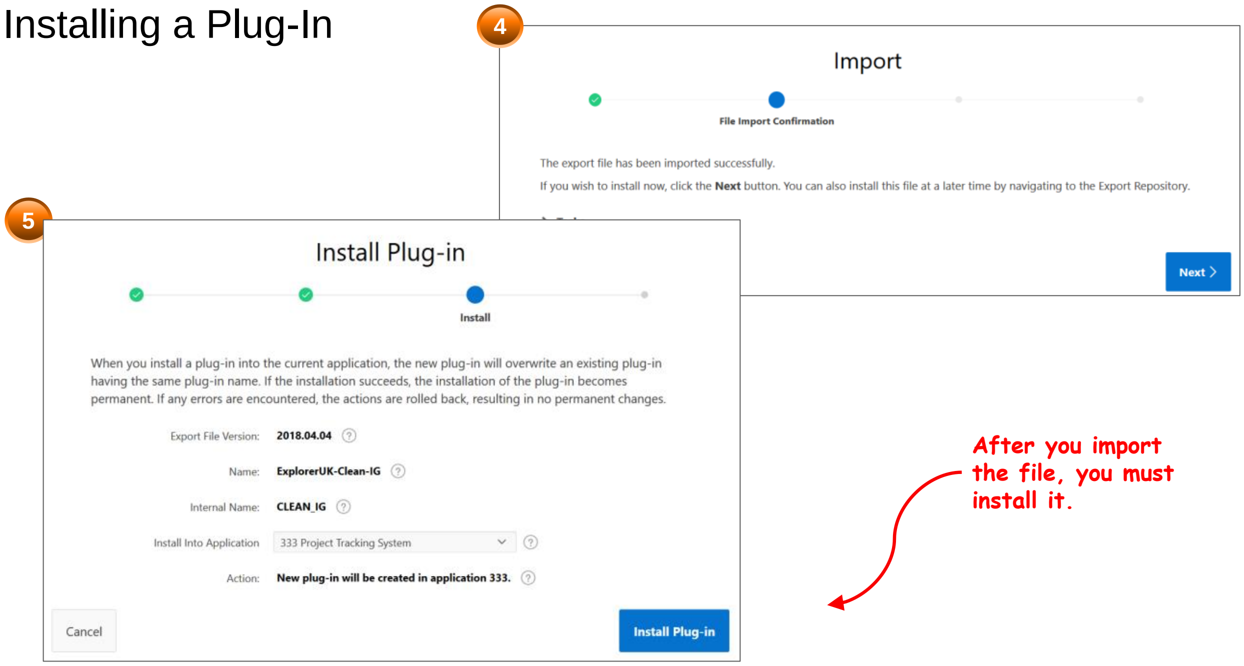
File Character Set: Unicode UTF-8

Import a SQL file that contains the plug-in definition.

To use a plug-in in your application, you import it under Shared Components. To import a plug-in, perform the following steps:

1. Navigate to the **Shared Components** page in your application. Under **Other Components,** select **Plug-ins** (screenshot 1).
2. Click **Import** (screenshot 2). You must first download the plug-in from the Plug-in Repository and save it locally in your system.
3. Browse and select your plug-in import file and click **Next** (screenshot 3).

Installing a Plug-In



Import

The export file has been imported successfully.

If you wish to install now, click the **Next** button. You can also install this file at a later time by navigating to the Export Repository.

Install Plug-in

When you install a plug-in into the current application, the new plug-in will overwrite an existing plug-in having the same plug-in name. If the installation succeeds, the installation of the plug-in becomes permanent. If any errors are encountered, the actions are rolled back, resulting in no permanent changes.

Export File Version: **2018.04.04**

Name: **ExplorerUK-Clean-IG**

Internal Name: **CLEAN_IG**

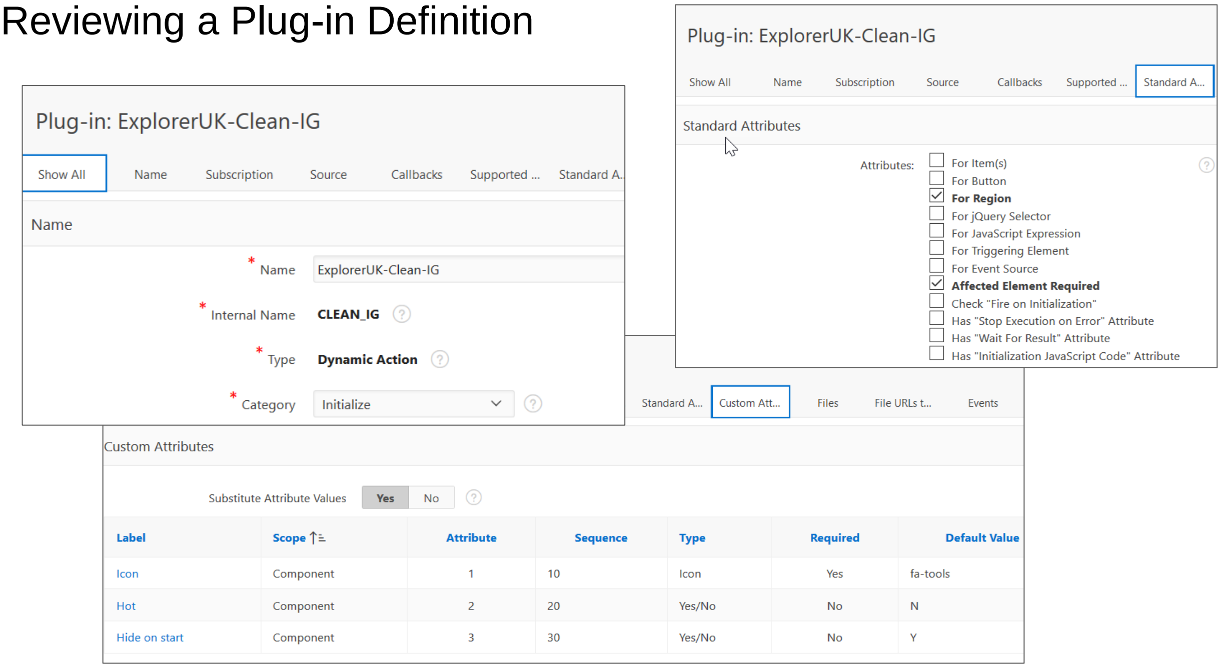
Install Into Application: 333 Project Tracking System

Action: **New plug-in will be created in application 333.**

After you import the file, you must install it.

4. After the file is imported, click **Next** to install it.
5. Select the application that you want to install the plug-in into and click **Install Plug-in.**

Reviewing a Plug-in Definition



Plug-in: ExplorerUK-Clean-IG

Name

* Name: ExplorerUK-Clean-IG

* Internal Name: CLEAN_IG

* Type: Dynamic Action

* Category: Initialize

Plug-in: ExplorerUK-Clean-IG

Standard Attributes

Attributes:

- □ For Item(s)
- □ For Button
- ☑ **For Region**
- □ For jQuery Selector
- □ For JavaScript Expression
- □ For Triggering Element
- □ For Event Source
- ☑ **Affected Element Required**
- □ Check "Fire on Initialization"
- □ Has "Stop Execution on Error" Attribute
- □ Has "Wait For Result" Attribute
- □ Has "Initialization JavaScript Code" Attribute

Custom Attributes

Substitute Attribute Values: [✓ Yes] No

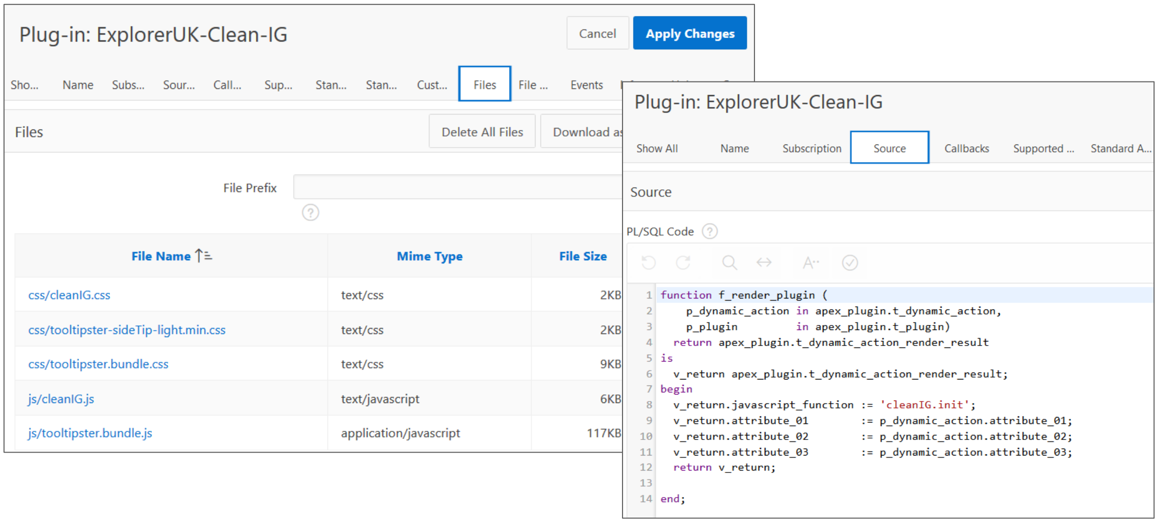
Label	Scope ↑	Attribute	Sequence	Type	Required	Default Value
Icon	Component	1	10	Icon	Yes	fa-tools
Hot	Component	2	20	Yes/No	No	N
Hide on start	Component	3	30	Yes/No	No	Y

If you want to provide your user with different experiences while viewing the interactive grid and while editing it, then this plug-in is a good option as it gives the user the choice to show and hide the toolbar.

After installing the plug-in, review the plug-in definition. To review the contents of a plug-in, navigate to **Shared Components,** then go to **Plug-ins** for your application and click the plug-in icon. Some of the sections that appear in the definition include:

- **Name:** Provides a name for the plug-in, an internal name, and type of plug-in it represents. As you can see in the screenshot, the *ExplorerUK_Clean-IG* plug-in is a Dynamic Action type plug-in.
- **Standard Attributes:** Contains a list of attributes that apply to this plug-in
- **Custom Attributes:** Prompts the developer for additional data in the Builder when the plug-in is used

Reviewing a Plug-in Definition



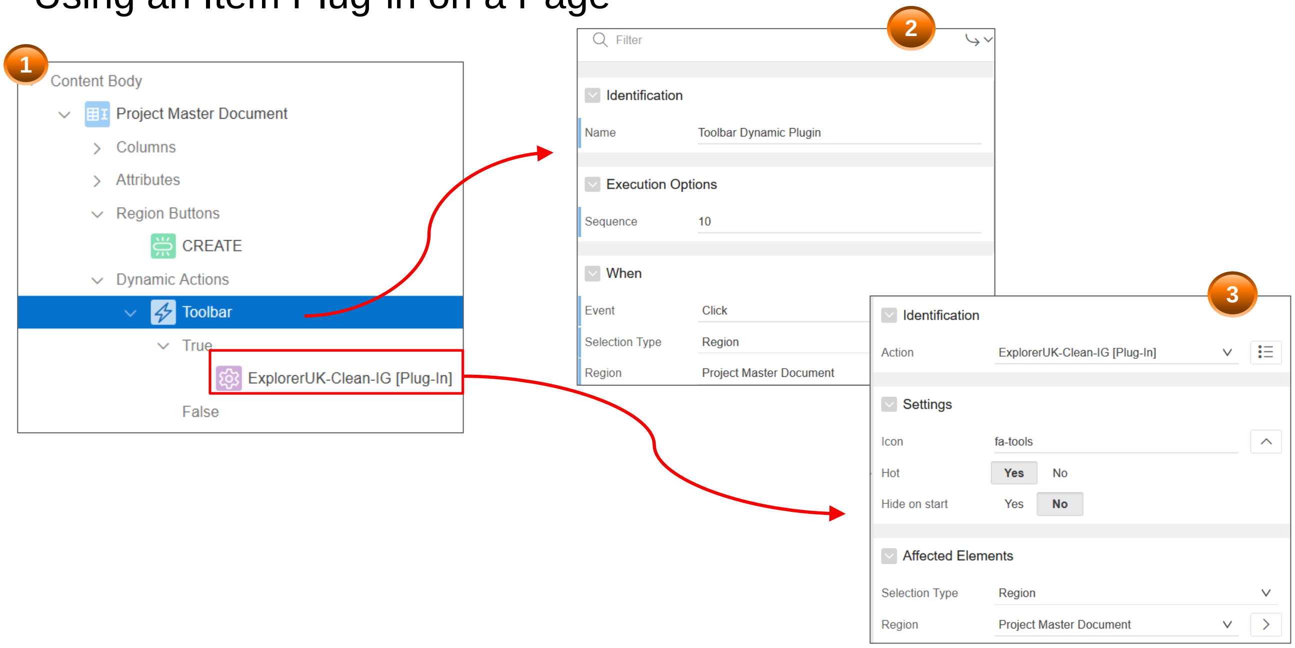
File Name	Mime Type	File Size
css/cleanIG.css	text/css	2KB
css/tooltipster-sideTip-light.min.css	text/css	2KB
css/tooltipster.bundle.css	text/css	9KB
js/cleanIG.js	text/javascript	6KB
js/tooltipster.bundle.js	application/javascript	117KB

```
function f_render_plugin (
    p_dynamic_action in apex_plugin.t_dynamic_action,
    p_plugin         in apex_plugin.t_plugin)
  return apex_plugin.t_dynamic_action_render_result
is
  v_return apex_plugin.t_dynamic_action_render_result;
begin
  v_return.javascript_function := 'cleanIG.init';
  v_return.attribute_01        := p_dynamic_action.attribute_01;
  v_return.attribute_02        := p_dynamic_action.attribute_02;
  v_return.attribute_03        := p_dynamic_action.attribute_03;
  return v_return;

end;
```

- **Files:** Displays the images, style sheet, and JavaScript files needed for this plug-in to run successfully
- **Source:** Provides the PL/SQL code that is needed to run so that the plug-in will function properly

Using an Item Plug-in on a Page



Content Body
- Project Master Document
  - Columns
  - Attributes
  - Region Buttons
    - CREATE
  - Dynamic Actions
    - Toolbar
      - True
        - ExplorerUK-Clean-IG [Plug-In]
      - False

Identification

Name: Toolbar Dynamic Plugin

Execution Options

Sequence: 10

When

Event: Click

Selection Type: Region

Region: Project Master Document

Identification

Action: ExplorerUK-Clean-IG [Plug-In]

Settings

Icon: fa-tools

Hot: Yes

Hide on start: No

Affected Elements

Selection Type: Region

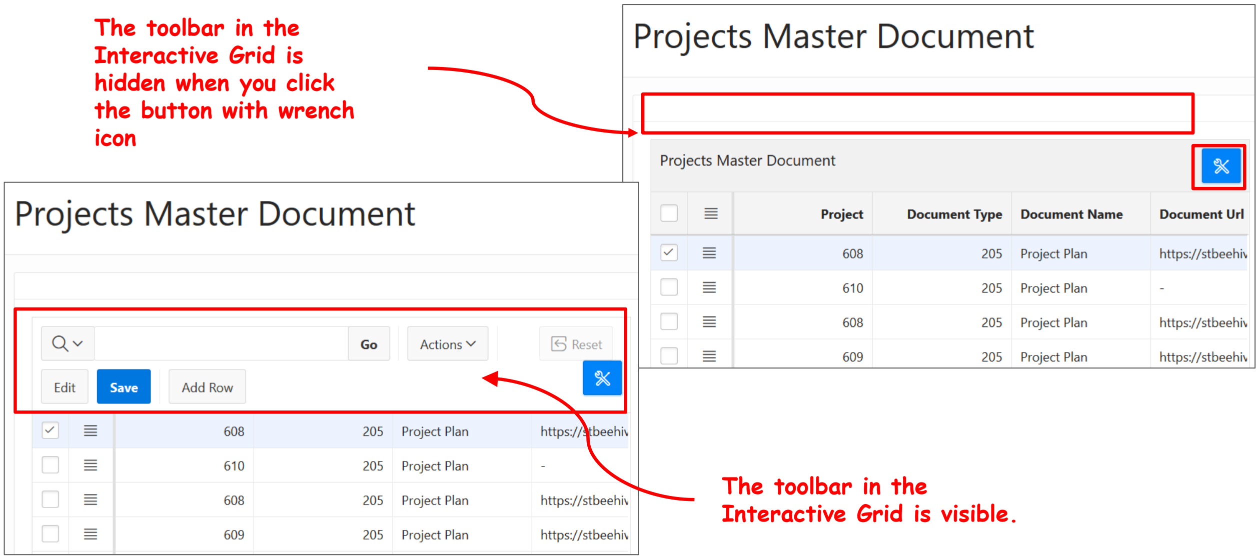
Region: Project Master Document

After the plug-in is made available to the application, you can use it. In this example, Steve uses the *ExplorerUK_Clean_IG [Plug-in]* on the *Project Master Document* interactive grid in the PTS application. The plug-in shows and hides the interactive grid toolbar when the user clicks the plug-in button.

Perform the following steps to use the plug-in:

1. Open the PTS application and click **Page 7 Project Master Document**. The page opens in Page Designer.
2. Under **Dynamic Actions** tab, right-click **Events** and click **Create Dynamic Actions.**
3. In the Property Editor on the right pane, enter the following:
   - **Name:** Toolbar Dynamic Plugin
   - Under **When, Event:** Select **Click.** This triggers the dynamic action plug-in when the user clicks the button
   - **Selection Type:** Region
   - **Region:** Project Master Document
4. On the left pane, expand **True** under the Dynamic Action and click **Show.** On the right pane, enter the following:
   - Under **Identification, Action:** Select **ExplorerUK_Clean_IG [Plug-in]**
   - Under **Settings, Icon:** Select **fa-tools**
   - **Hot:** Yes. Selecting yes for this option gives a distinct look to the button.
   - **Hide on Start:** No

Using an Item Plug-in on a Page



The toolbar in the Interactive Grid is hidden when you click the button with wrench icon

The toolbar in the Interactive Grid is visible.

Projects Master Document

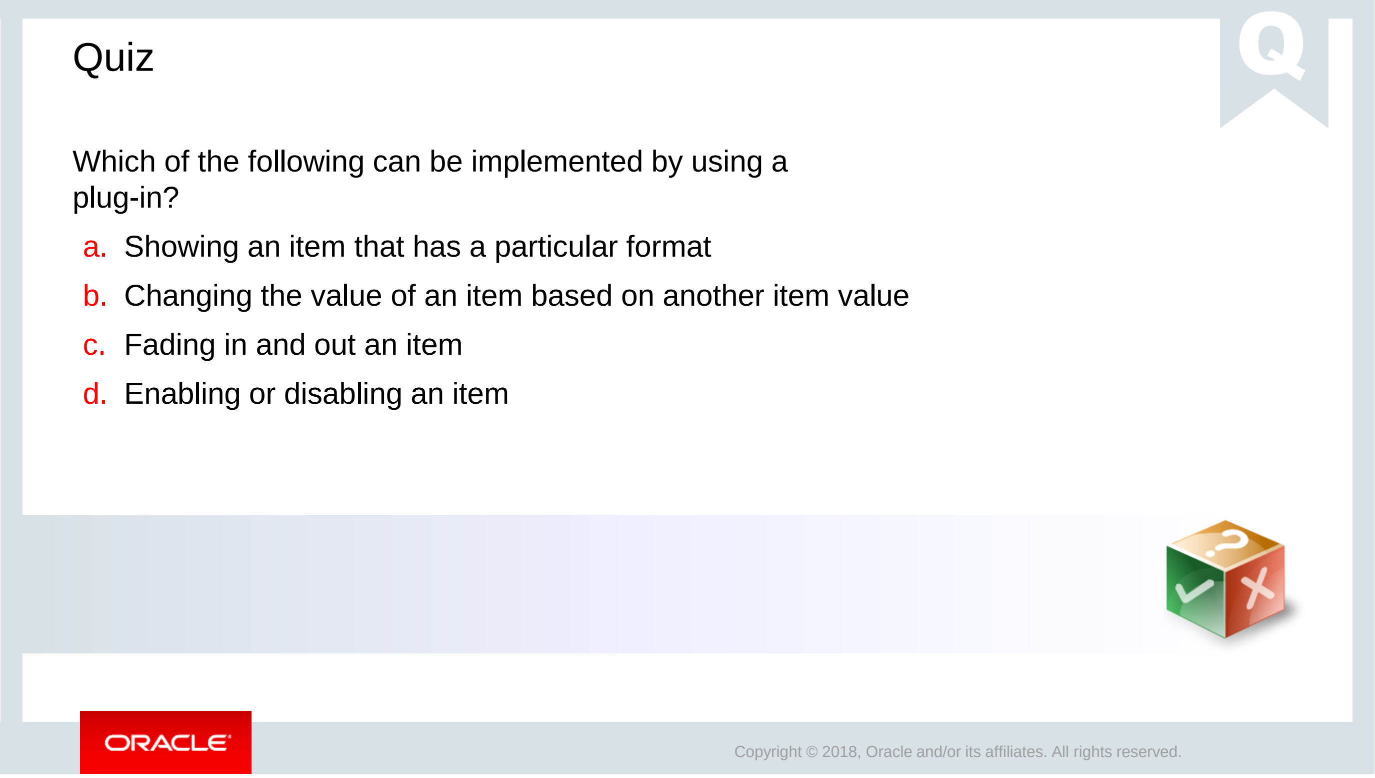
Projects Master Document

		Project	Document Type	Document Name	Document Url
☑	≣	608	205	Project Plan	https://stbeehiv
□	≣	610	205	Project Plan	-
□	≣	608	205	Project Plan	https://stbeehiv
□	≣	609	205	Project Plan	https://stbeehiv

Projects Master Document

☑	≣	608	205	Project Plan	https://stbeehiv
□	≣	610	205	Project Plan	-
□	≣	608	205	Project Plan	https://stbeehiv
□	≣	609	205	Project Plan	https://stbeehiv

5. Click **Save** and **Run** to load the *Project Master Document* interactive grid. You will observe that a new button with the wrench icon is displayed along the toolbar. Click the button with the wrench icon. The toolbar is hidden. Click the wrench icon once again, and the interactive grid toolbar is displayed.



Quiz

Which of the following can be implemented by using a plug-in?

a. Showing an item that has a particular format

b. Changing the value of an item based on another item value

c. Fading in and out an item

d. Enabling or disabling an item

ORACLE

Answer: a, c

b and d are dynamic actions.

Additional Plug-in Examples

- Adding a rating to your Employees form page
  - Add the Modern Star Rating plug-in on the Employees form in the PTS application.
- Displaying notification message When an Item Is Clicked
  - Create a dynamic action that uses the notification plug-in on the *Project Members* form attached to the *Project Status Report.* When you enter **Y** in the **Milestone YN** field while creating an action item in the project, then a notification is displayed asking you to enter the milestone date without fail. This is because the **Milestone Date** field becomes a mandatory field if the value for **Milestone YN = Y**

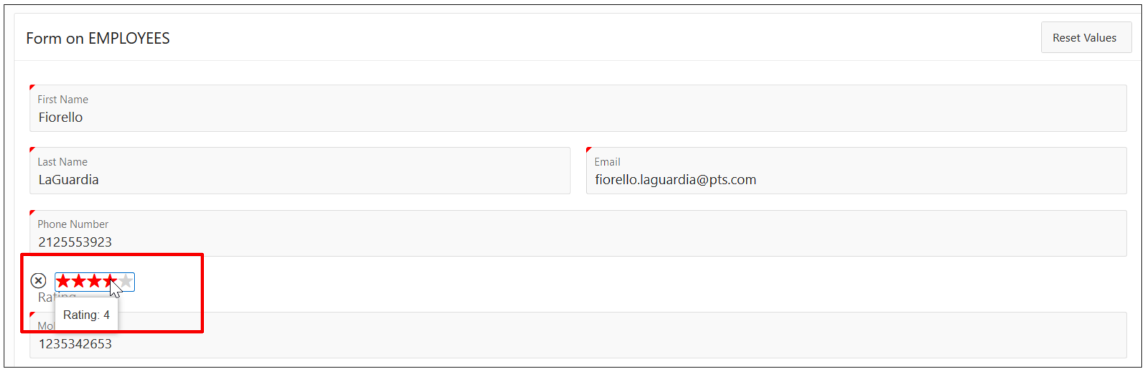
There are several plug-ins in the Plug-in Repository that you can access from https://www.oracle.com/technetwork/developer-tools/apex/application-express/apex-plug-ins-182042.html. You examine some additional examples provided in the slide.

To find out more about plug-ins, see the *Application Express User’s Guide*.

Adding the Modern Star Rating Item

To add the Modern Star Rating Item plug-in:

- Add the Rating column in the EMPLOYEES table
- Import the Modern Star Rating Plug-in into the PTS application
- Implement the Modern Star Rating plug-in on the EMPLOYEES form



Form on EMPLOYEES

First Name
Fiorello

Last Name
LaGuardia

Email
fiorello.laguardia@pts.com

Phone Number
2125553923

★★★★☆
Rating: 4

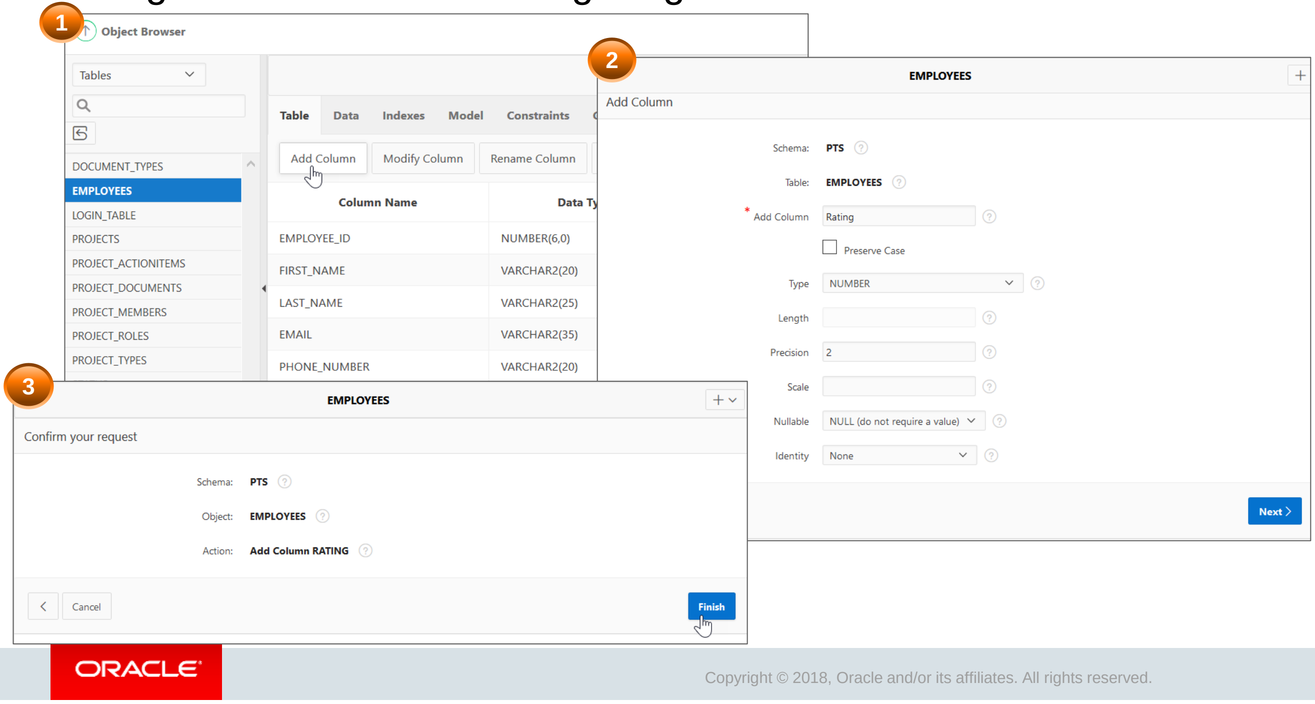
1235342653

Steve decides to add a functionality to rate employees in the PTS application. He implements it on the Employees form.

To implement the Modern Star Rating plug-in, you must execute the following tasks:

- First, add a column in the `EMPLOYEES` table. This column contains the rating data for each employee. Steve adds the column named `RATING`.
- Then import and install the *Modern Star Rating plug-in* into the application in which you want to implement it.
- Last, implement it on the specific page in the application. Steve adds the plug-in on the `EMPLOYEES` form in the PTS application.

Adding the Modern Star Rating Plug-in—Add Column



Tables
DOCUMENT_TYPES
EMPLOYEES
LOGIN_TABLE
PROJECTS
PROJECT_ACTIONITEMS
PROJECT_DOCUMENTS
PROJECT_MEMBERS
PROJECT_ROLES
PROJECT_TYPES

Column Name	Data Ty
EMPLOYEE_ID	NUMBER(6,0)
FIRST_NAME	VARCHAR2(20)
LAST_NAME	VARCHAR2(25)
EMAIL	VARCHAR2(35)
PHONE_NUMBER	VARCHAR2(20)

EMPLOYEES

Add Column

Schema: PTS

Table: EMPLOYEES

* Add Column: Rating

□ Preserve Case

Type: NUMBER

Length:

Precision: 2

Scale:

Nullable: NULL (do not require a value)

Identity: None

EMPLOYEES

Confirm your request

Schema: PTS

Object: EMPLOYEES

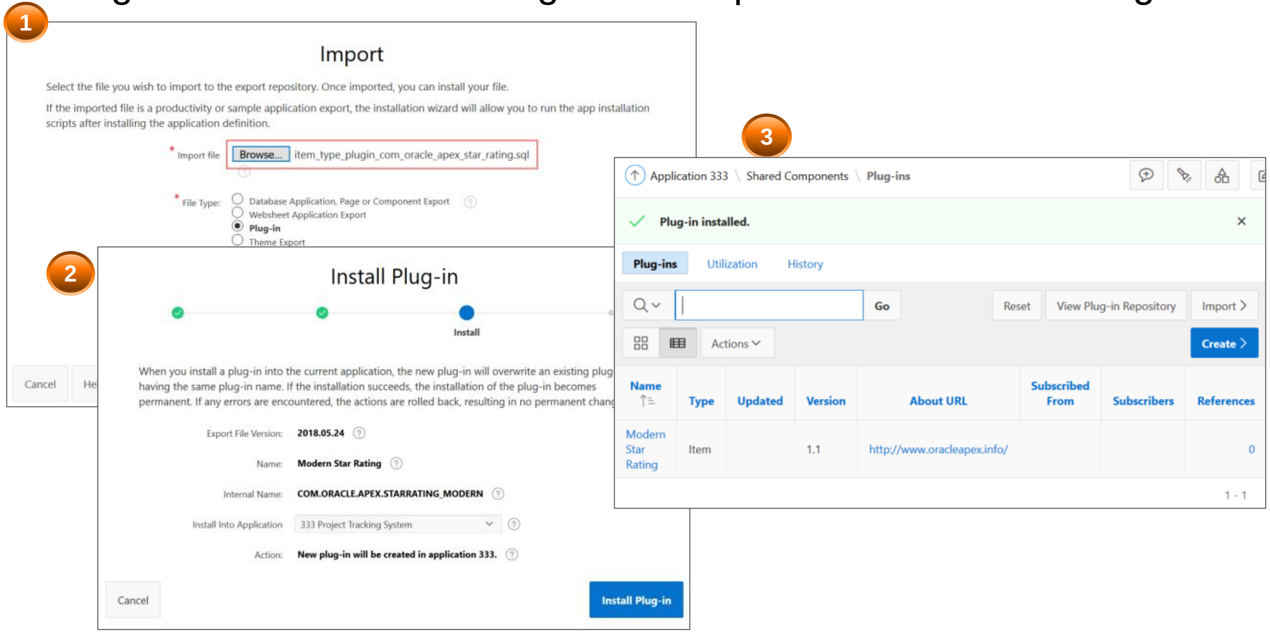
Action: Add Column RATING

ORACLE

Steve adds the RATING column in the EMPLOYEES table. This is the first step in implementing the Modern Star Rating plug-in in an application. To do so:

1. Go to **SQL Developer** and then **Object Browser**.
2. Click **PTS** under schema and click EMPLOYEES.
3. In the Detail pane (right pane), click the **Add Column** button and define the following:
   - **Add Column:** Enter RATING
   - **Type:** Select **Number**
   - **Precision:** Enter **2**
   - Click **Next >.**
4. Click **Finish.** The RATING column is added to the EMPLOYEES table.

Adding the Modern Star Rating Item—Import and Install the Plug-in



Import

Select the file you wish to import to the export repository. Once imported, you can install your file.

If the imported file is a productivity or sample application export, the installation wizard will allow you to run the app installation scripts after installing the application definition.

* Import file: item_type_plugin_com_oracle_apex_star_rating.sql

* File Type:
○ Database Application, Page or Component Export
○ Websheet Application Export
● Plug-in
○ Theme Export

Install Plug-in

When you install a plug-in into the current application, the new plug-in will overwrite an existing plug having the same plug-in name. If the installation succeeds, the installation of the plug-in becomes permanent. If any errors are encountered, the actions are rolled back, resulting in no permanent chang

Export File Version: 2018.05.24

Name: Modern Star Rating

Internal Name: COM.ORACLE.APEX.STARRATING_MODERN

Install Into Application: 333 Project Tracking System

Action: New plug-in will be created in application 333.

Application 333 \ Shared Components \ Plug-ins

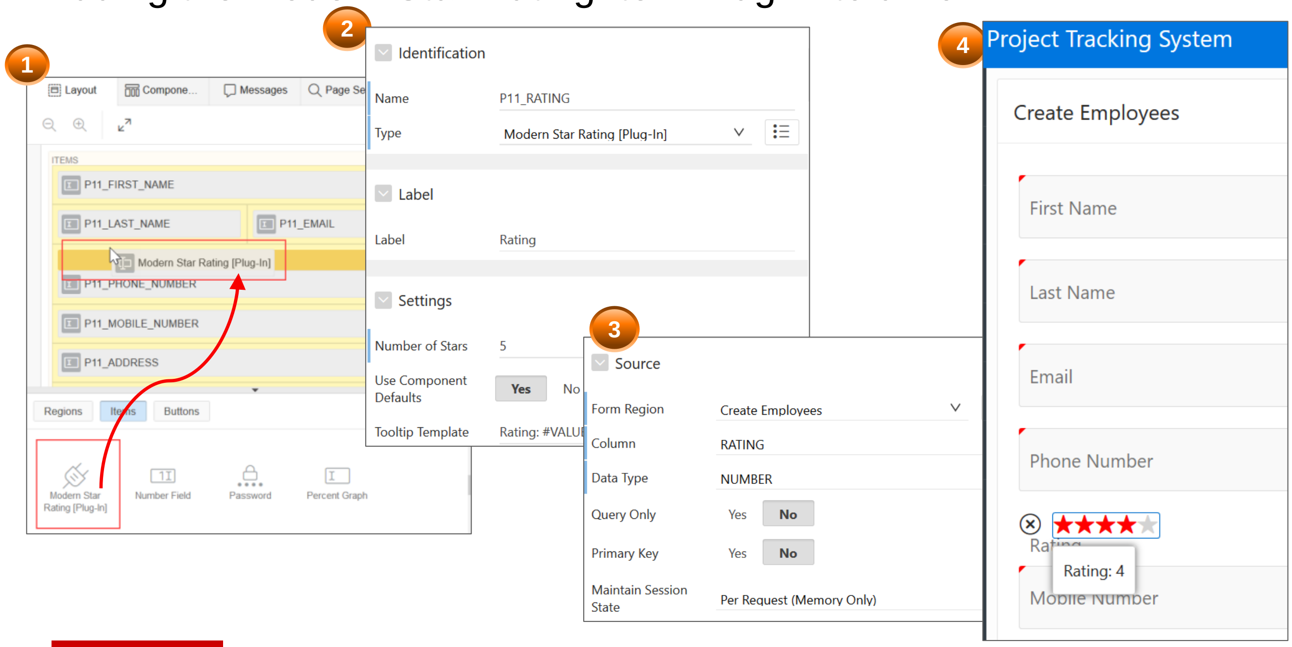
✓ Plug-in installed.

Name	Type	Updated	Version	About URL	Subscribed From	Subscribers	References
Modern Star Rating	Item		1.1	http://www.oracleapex.info/			0

This is the second step in implementing the Modern Star Rating plug-in—to import and install it in the PTS application. To do so:

1. Go to the PTS application home page and navigate to **Shared Components.**
2. In Shared Components, click **Plug-ins** under Other Components.
3. Click **Import**, select the option Plug-in and browse to the location where the Modern Star Rating item plug-in is available, and click **Next.** You can also select the file from `/home/oracle/labs/demos/files/item_type_plugin_com_oracle_apex_starrating_modern.sql`
4. In the Install Plug-in dialog, select **333 Project Tracking System** for Install Into Application field and click **Install Plug-in.**
5. In Component Settings, enter the name *Clear Rating* in the **Clear Tooltip** field and click **Apply Changes.** The message Plug-in installed is displayed.

Adding the Modern Star Rating Item Plug-in to a Form



Identification

Name	P11_RATING
Type	Modern Star Rating [Plug-In]

Label

Label	Rating

Settings

Number of Stars	5
Use Component Defaults	Yes No
Tooltip Template	Rating: #VALUE

Source

Form Region	Create Employees
Column	RATING
Data Type	NUMBER
Query Only	Yes No
Primary Key	Yes No
Maintain Session State	Per Request (Memory Only)

Project Tracking System

Create Employees

First Name

Last Name

Email

Phone Number

★★★★☆

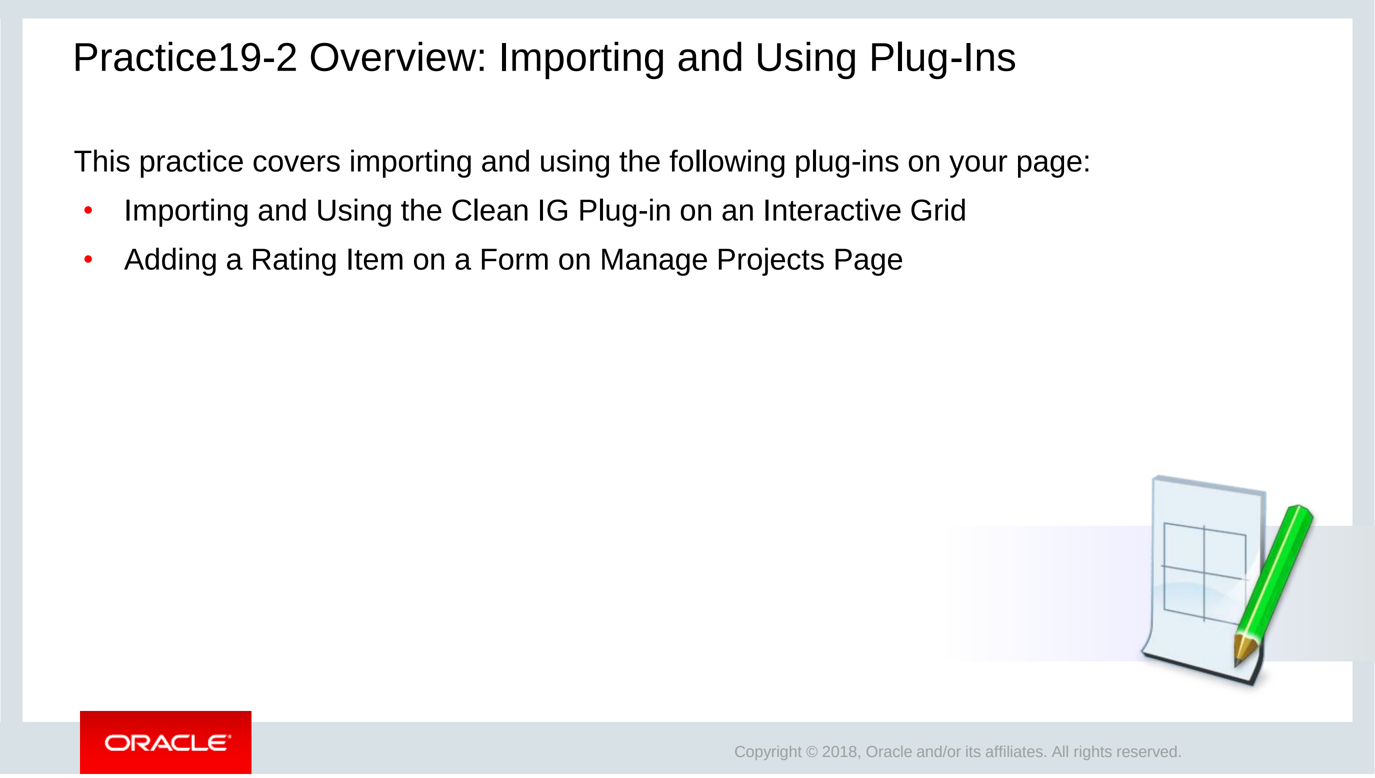
Rating

Rating: 4

Mobile Number

This is the last step in adding the Modern Star Rating plug-in, where Steve adds the Modern Star Rating plug-in on the EMPLOYEES form.

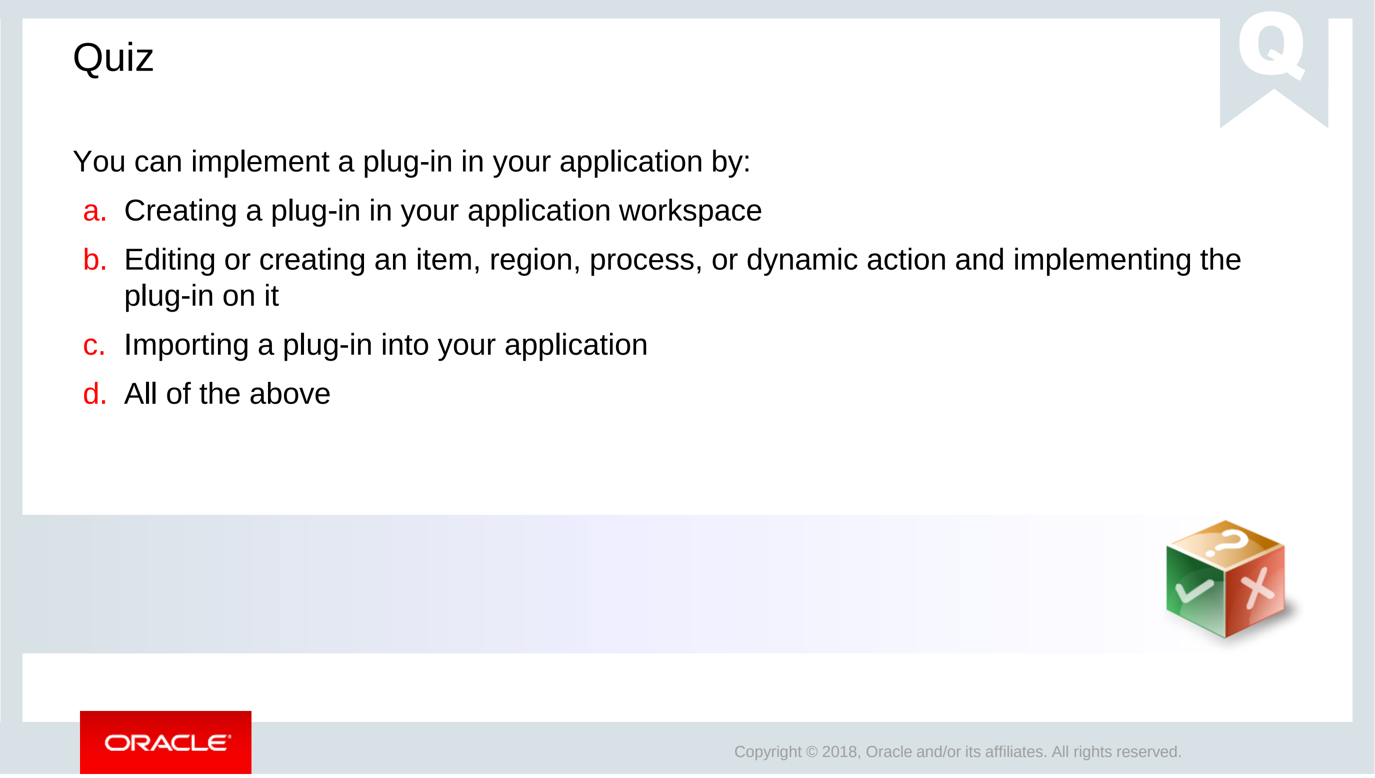
1. Open page 11 – Create Employees page in Page Designer.
2. In the Gallery at the bottom of the central pane, click the **Items** tab and locate the Modern Star Rating [Plug-In]. Select the **Modern Star Rating [Plug-In]** and then drag and drop it to after P11_PHONE_NUMBER in the Layout pane, as shown in screenshot 1.
3. In the Rendering tab, verify the newly added plug-in is selected. In the Property Editor, edit the following (as shown in screenshot 2 and 3):
   - **Identification** > **Name:** Enter P11_RATING
   - **Settings** > **Number of Stars:** Enter 5
   - **Appearance** > **Template:** Select Optional – Above
   - **Source** > **Type:** Select **Database Column.**
   - **Database Column:** Select RATING .This is the column you created on the EMPLOYEES table. You must select a database column in which the entry for employee rating (when the user rates the employees) is updated.
4. Click **Save** and **Run** to load Create Employees form, as shown in screenshot 4.



Practice19-2 Overview: Importing and Using Plug-Ins

This practice covers importing and using the following plug-ins on your page:

- Importing and Using the Clean IG Plug-in on an Interactive Grid
- Adding a Rating Item on a Form on Manage Projects Page



Quiz

You can implement a plug-in in your application by:

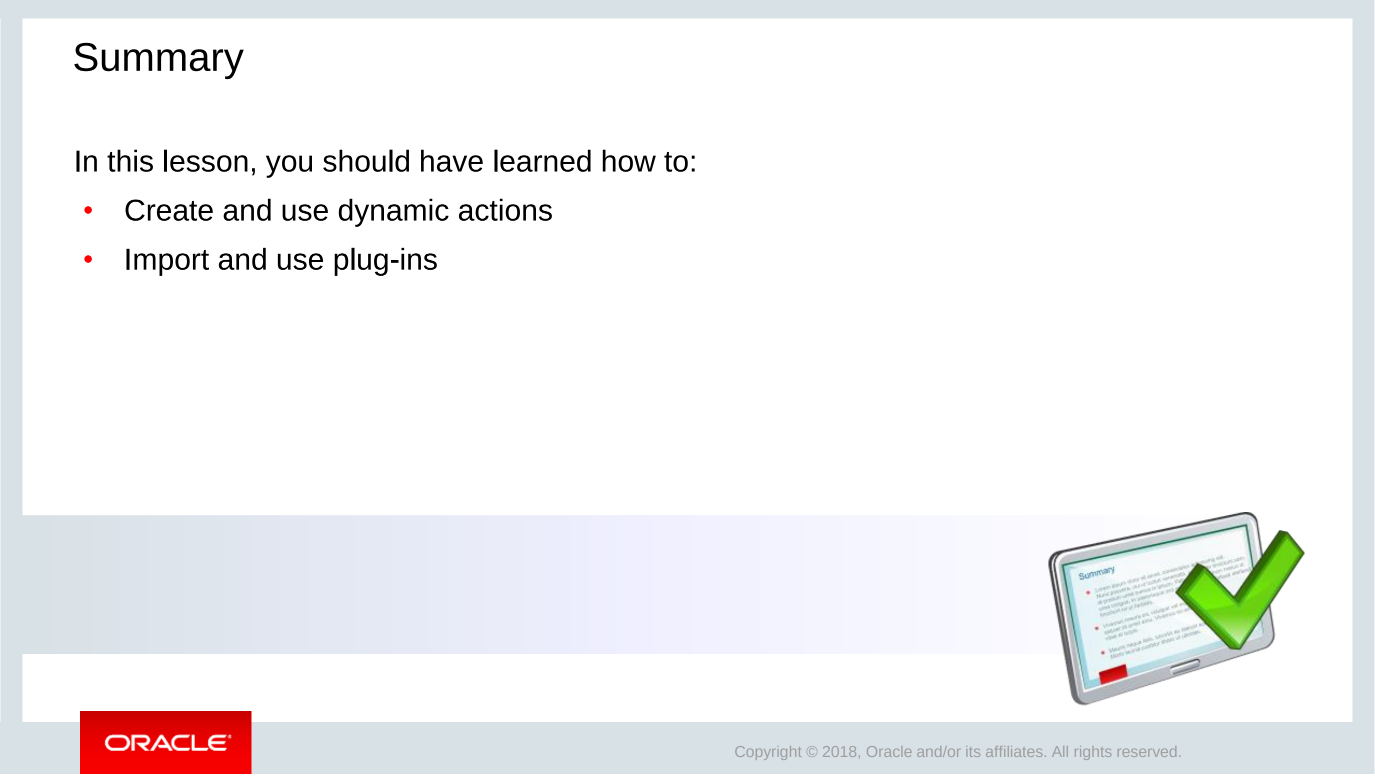
a. Creating a plug-in in your application workspace

b. Editing or creating an item, region, process, or dynamic action and implementing the plug-in on it

c. Importing a plug-in into your application

d. All of the above

Answer: d

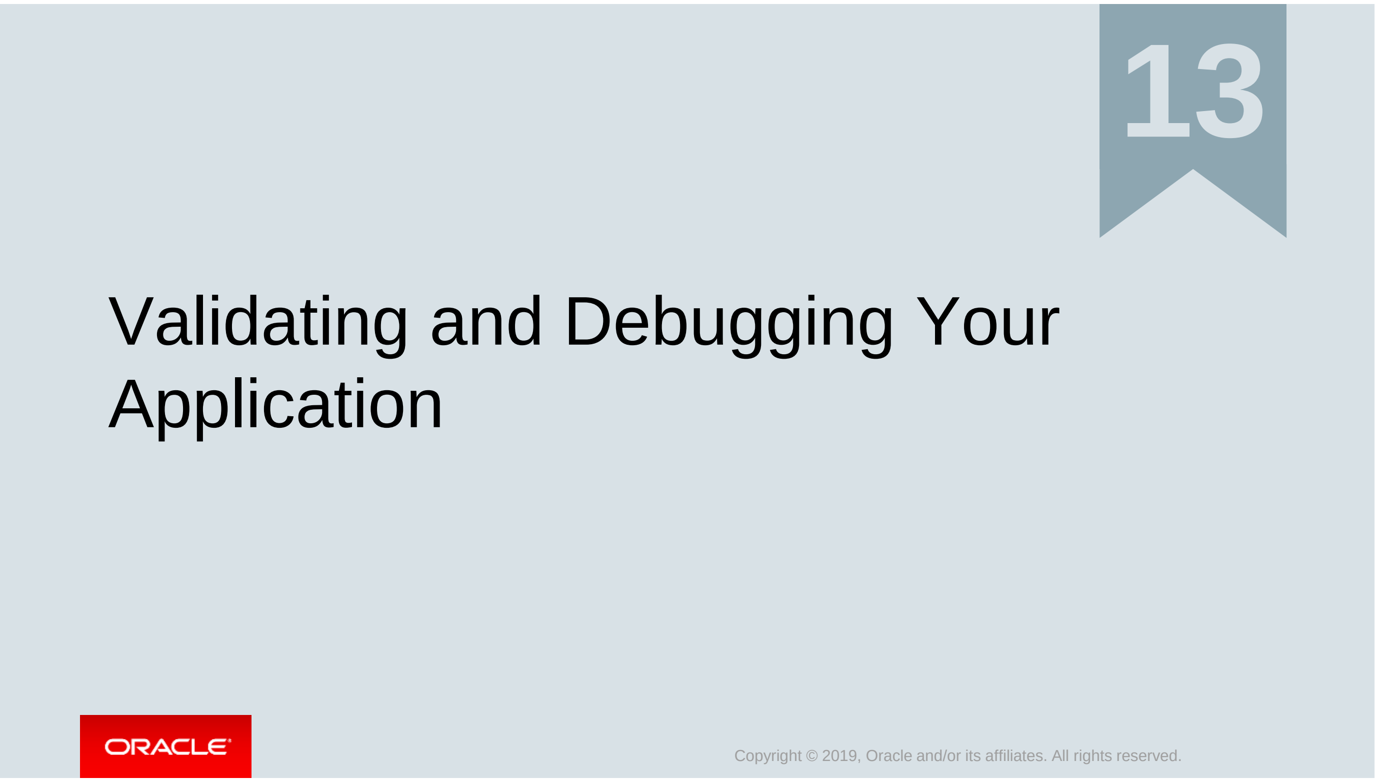


Summary

In this lesson, you should have learned how to:

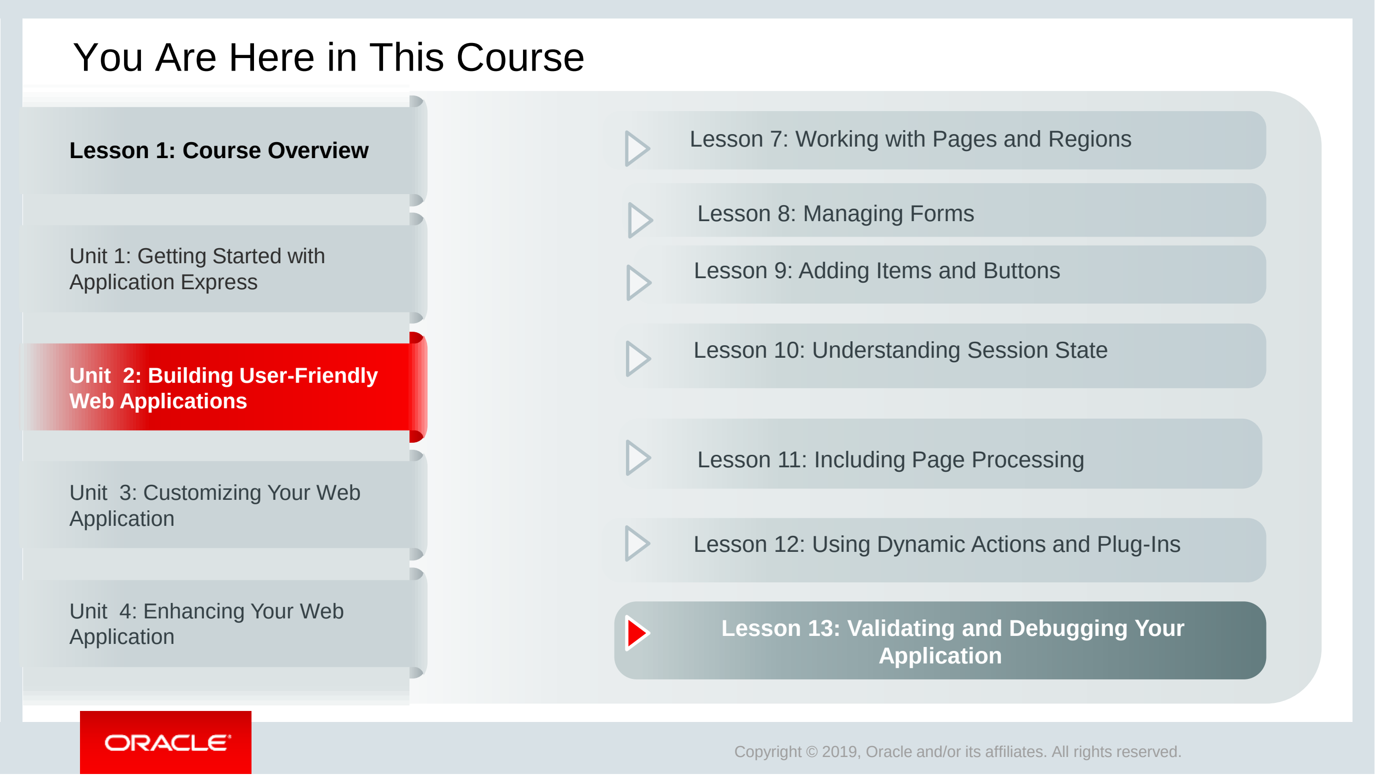
- Create and use dynamic actions
- Import and use plug-ins

In this lesson, you should have learned how to create and use dynamic actions and plug-ins in your application.



13

Validating and Debugging Your Application

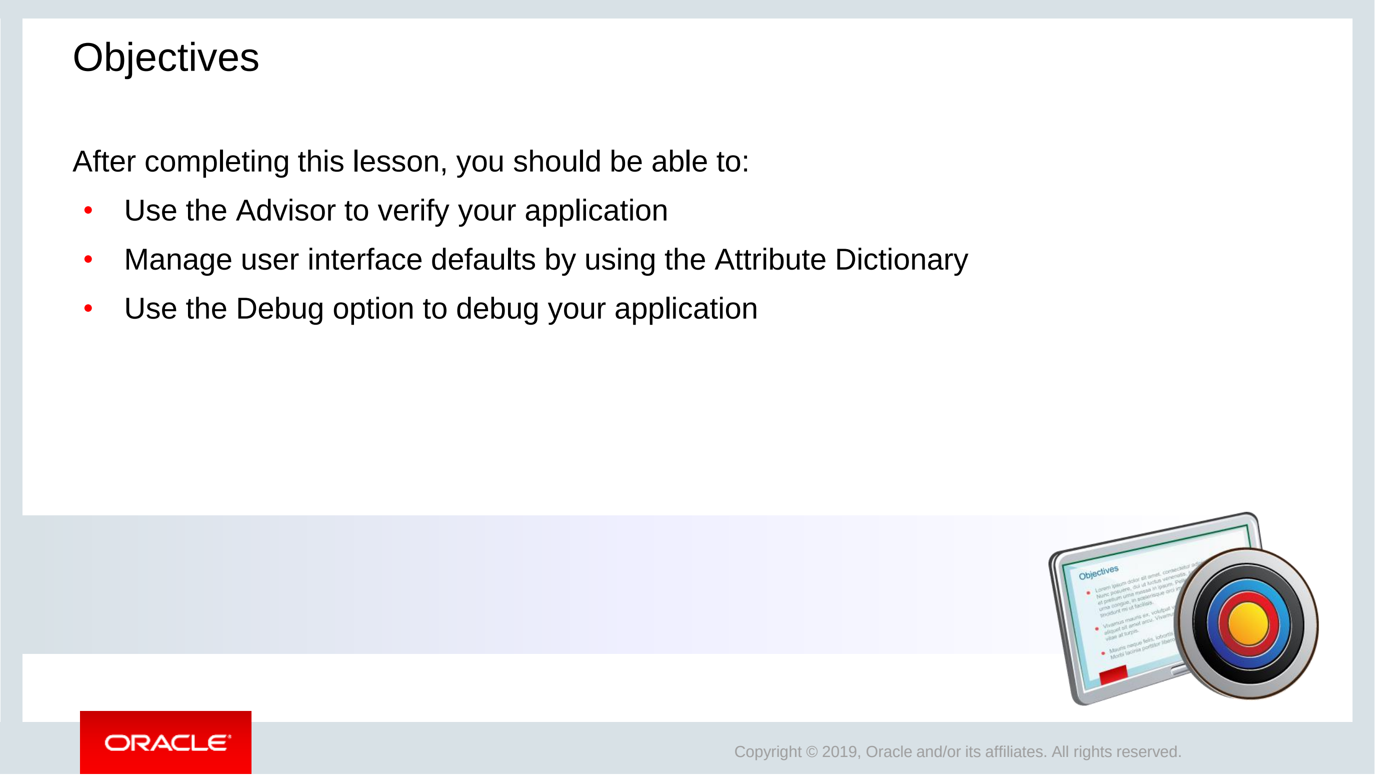


You Are Here in This Course

- **Lesson 1: Course Overview**
- Unit 1: Getting Started with Application Express
- **Unit 2: Building User-Friendly Web Applications**
- Unit 3: Customizing Your Web Application
- Unit 4: Enhancing Your Web Application

- Lesson 7: Working with Pages and Regions
- Lesson 8: Managing Forms
- Lesson 9: Adding Items and Buttons
- Lesson 10: Understanding Session State
- Lesson 11: Including Page Processing
- Lesson 12: Using Dynamic Actions and Plug-Ins
- **Lesson 13: Validating and Debugging Your Application**

This slide is a graphical depiction of the course, particularly highlighting Unit 2 – Lesson 13, which is dealt with in these slides.



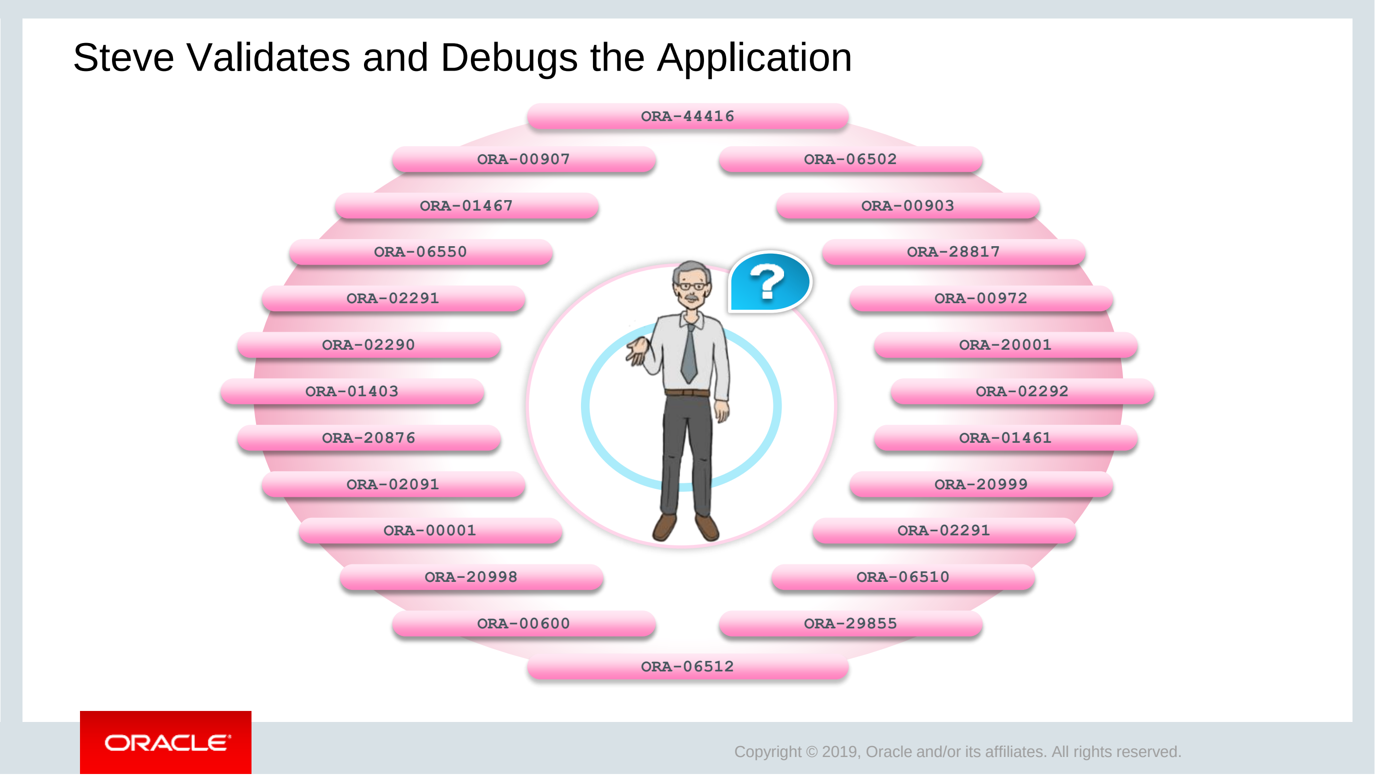
Objectives

After completing this lesson, you should be able to:

- Use the Advisor to verify your application
- Manage user interface defaults by using the Attribute Dictionary
- Use the Debug option to debug your application

ORACLE

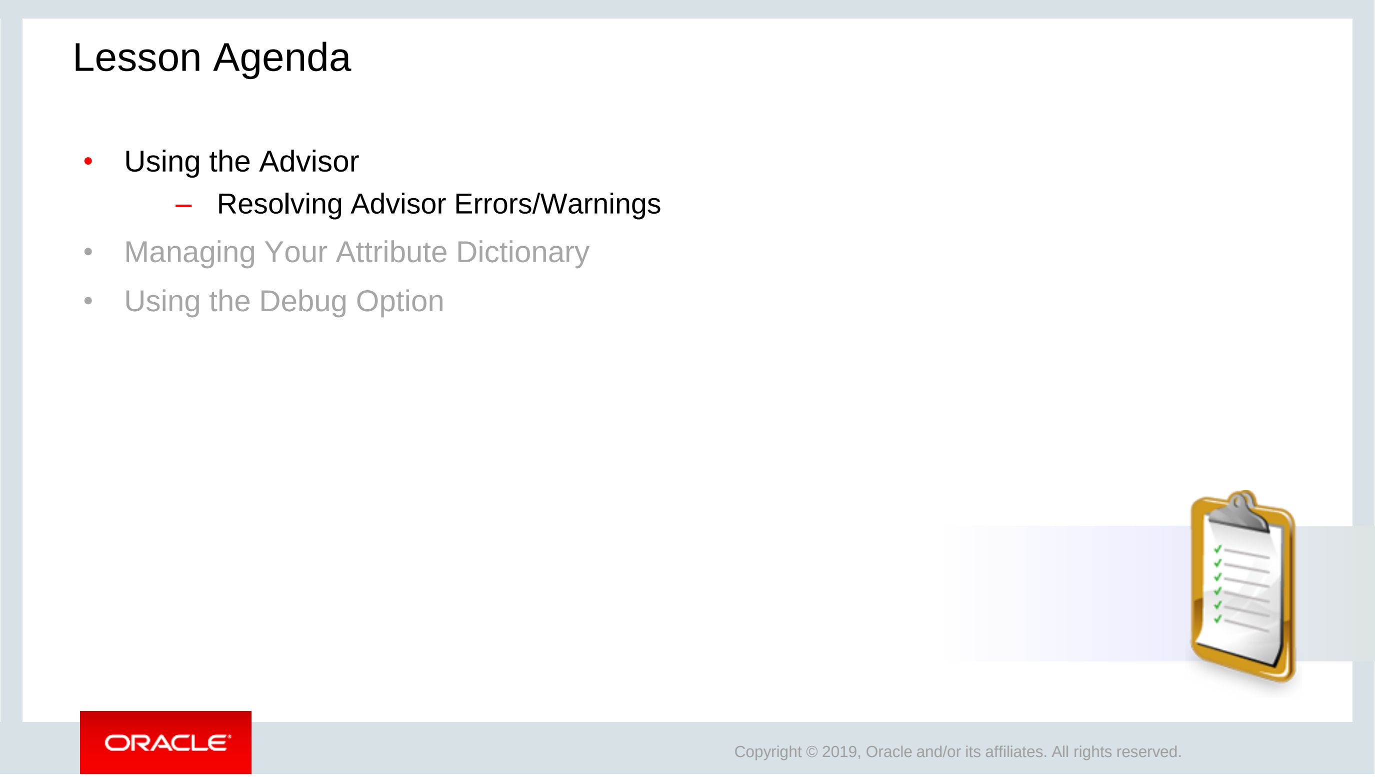
In this lesson, you learn how to use the Advisor to verify your application, manage user interface defaults by using Attribute Dictionary, and use the Debug option to view the debug messages.



Steve Validates and Debugs the Application

Steve has been trying his hands on Oracle Application Express and has explored the various functionalities available. He is in the final stages of developing the *PTS* application. While he is sure the application meets all the requirements that were specified by Stella, he wants to test and debug the application to ensure it runs as expected.

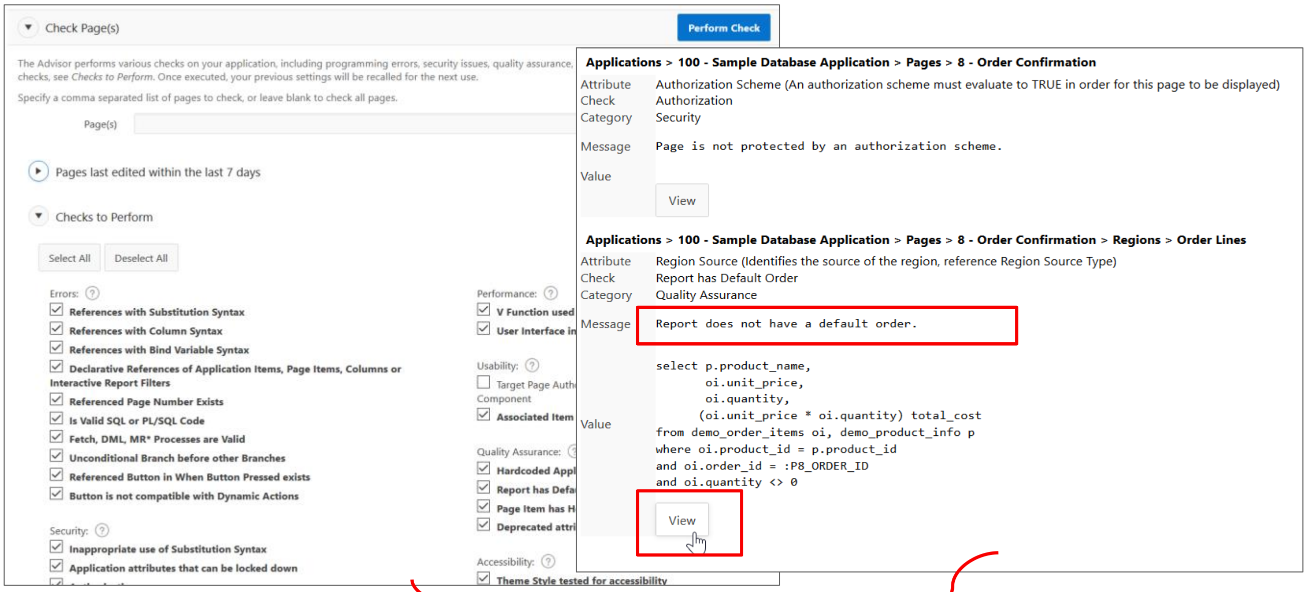
He wants to first test on the *Sample Database Application* (which we installed in the lesson titled “Oracle Application Express: Introduction”), as it is a fully functional and editable application, and later switch to *PTS* application to debug his application.



Lesson Agenda

- Using the Advisor
  - Resolving Advisor Errors/Warnings
- Managing Your Attribute Dictionary
- Using the Debug Option

Using the Advisor



▼ Check Page(s)

Perform Check

The Advisor performs various checks on your application, including programming errors, security issues, quality assurance, checks, see *Checks to Perform*. Once executed, your previous settings will be recalled for the next use.

Specify a comma separated list of pages to check, or leave blank to check all pages.

Page(s)

▶ Pages last edited within the last 7 days

▼ Checks to Perform

Select All | Deselect All

Errors: ?

- ☑ **References with Substitution Syntax**
- ☑ **References with Column Syntax**
- ☑ **References with Bind Variable Syntax**
- ☑ **Declarative References of Application Items, Page Items, Columns or Interactive Report Filters**
- ☑ **Referenced Page Number Exists**
- ☑ **Is Valid SQL or PL/SQL Code**
- ☑ **Fetch, DML, MR* Processes are Valid**
- ☑ **Unconditional Branch before other Branches**
- ☑ **Referenced Button in When Button Pressed exists**
- ☑ **Button is not compatible with Dynamic Actions**

Security: ?

- ☑ **Inappropriate use of Substitution Syntax**
- ☑ **Application attributes that can be locked down**

Performance: ?

- ☑ **V Function used**
- ☑ **User Interface in**

Usability: ?

- □ Target Page Auth Component
- ☑ **Associated Item**

Quality Assurance: ?

- ☑ **Hardcoded Appl**
- ☑ **Report has Defa**
- ☑ **Page Item has H**
- ☑ **Deprecated attri**

Accessibility: ?

- ☑ **Theme Style tested for accessibility**

Applications > 100 - Sample Database Application > Pages > 8 - Order Confirmation

Attribute	Authorization Scheme (An authorization scheme must evaluate to TRUE in order for this page to be displayed)
Check	Authorization
Category	Security
Message	`Page is not protected by an authorization scheme.`
Value	View

Applications > 100 - Sample Database Application > Pages > 8 - Order Confirmation > Regions > Order Lines

Attribute	Region Source (Identifies the source of the region, reference Region Source Type)
Check	Report has Default Order
Category	Quality Assurance
Message	`Report does not have a default order.`
Value	

```
select p.product_name,
       oi.unit_price,
       oi.quantity,
      (oi.unit_price * oi.quantity) total_cost
from demo_order_items oi, demo_product_info p
where oi.product_id = p.product_id
and oi.order_id = :P8_ORDER_ID
and oi.quantity <> 0
```

View

List of issues displayed based on your selections.

Can you see the violations reported in the example above? We see that quality assurance violations were found by Oracle Application Express *Advisor* because the *Order Lines* Classic Report does not have a default order mentioned in the SQL query.

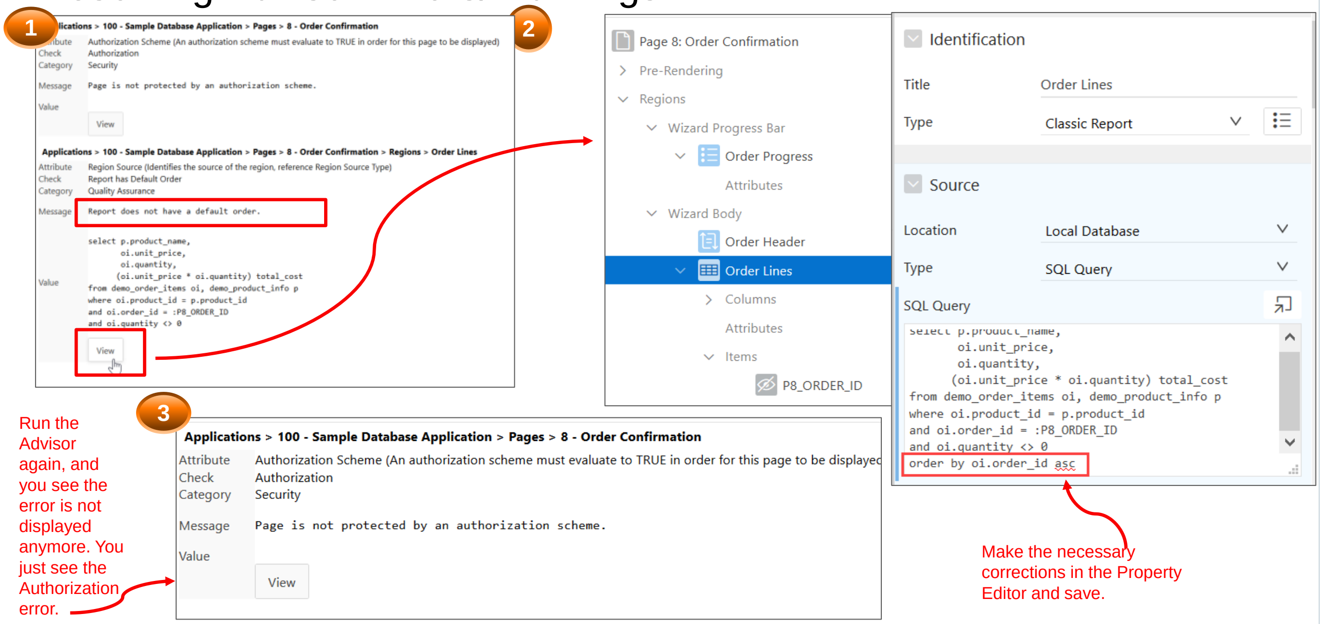
Now as you know, Steve wants to ensure that the *PTS* application runs without errors and meets Stella's requirements. To identify defects in the application, Steve can use the Oracle Application Express *Advisor*.

He can use the *Advisor* to check the integrity and quality of his application. So, before deploying his application, he can perform various sanity checks on his application like checks for errors, security issues, performance bottlenecks, quality assurance, and other best practices. He can click the **View** link for each violation to go to the page where he can correct the issue and then return to the *Advisor* to recheck.

In the next slide, let’s see how Steve runs the *Advisor* and resolves the above error.

Note that in the above example, Steve also gets an Authorization error in the *Order Confirmation* page. This is because the page is not protected by any authorization scheme. However, we will discuss about authorization in our lesson on *Enabling Security* in Unit 3.

Resolving Advisor Errors/Warnings



Applications > 100 - Sample Database Application > Pages > 8 - Order Confirmation

Attribute	Authorization Scheme (An authorization scheme must evaluate to TRUE in order for this page to be displayed)
Check	Authorization
Category	Security
Message	Page is not protected by an authorization scheme.
Value	

Applications > 100 - Sample Database Application > Pages > 8 - Order Confirmation > Regions > Order Lines

Attribute	Region Source (Identifies the source of the region, reference Region Source Type)
Check	Report has Default Order
Category	Quality Assurance
Message	Report does not have a default order.

Value

```
select p.product_name,
       oi.unit_price,
       oi.quantity,
      (oi.unit_price * oi.quantity) total_cost
from demo_order_items oi, demo_product_info p
where oi.product_id = p.product_id
and oi.order_id = :P8_ORDER_ID
and oi.quantity <> 0
```

Title: Order Lines
Type: Classic Report
Location: Local Database
Type: SQL Query
SQL Query

```
select p.product_name,
       oi.unit_price,
       oi.quantity,
      (oi.unit_price * oi.quantity) total_cost
from demo_order_items oi, demo_product_info p
where oi.product_id = p.product_id
and oi.order_id = :P8_ORDER_ID
and oi.quantity <> 0
order by oi.order_id asc
```

Run the Advisor again, and you see the error is not displayed anymore. You just see the Authorization error.

Applications > 100 - Sample Database Application > Pages > 8 - Order Confirmation

Attribute	Authorization Scheme (An authorization scheme must evaluate to TRUE in order for this page to be displayed
Check	Authorization
Category	Security
Message	Page is not protected by an authorization scheme.
Value	

Make the necessary corrections in the Property Editor and save.

As you saw in the previous slide, Steve had a quality assurance issue as the *Order Lines* Report does not have a default order mentioned in the SQL query. He now wants to address this error by sorting the Orders by the Order ID. To run the Advisor on an entire application, perform the following steps:

1. Navigate to your application.
2. Click **Utilities**.
3. Click **Advisor**.

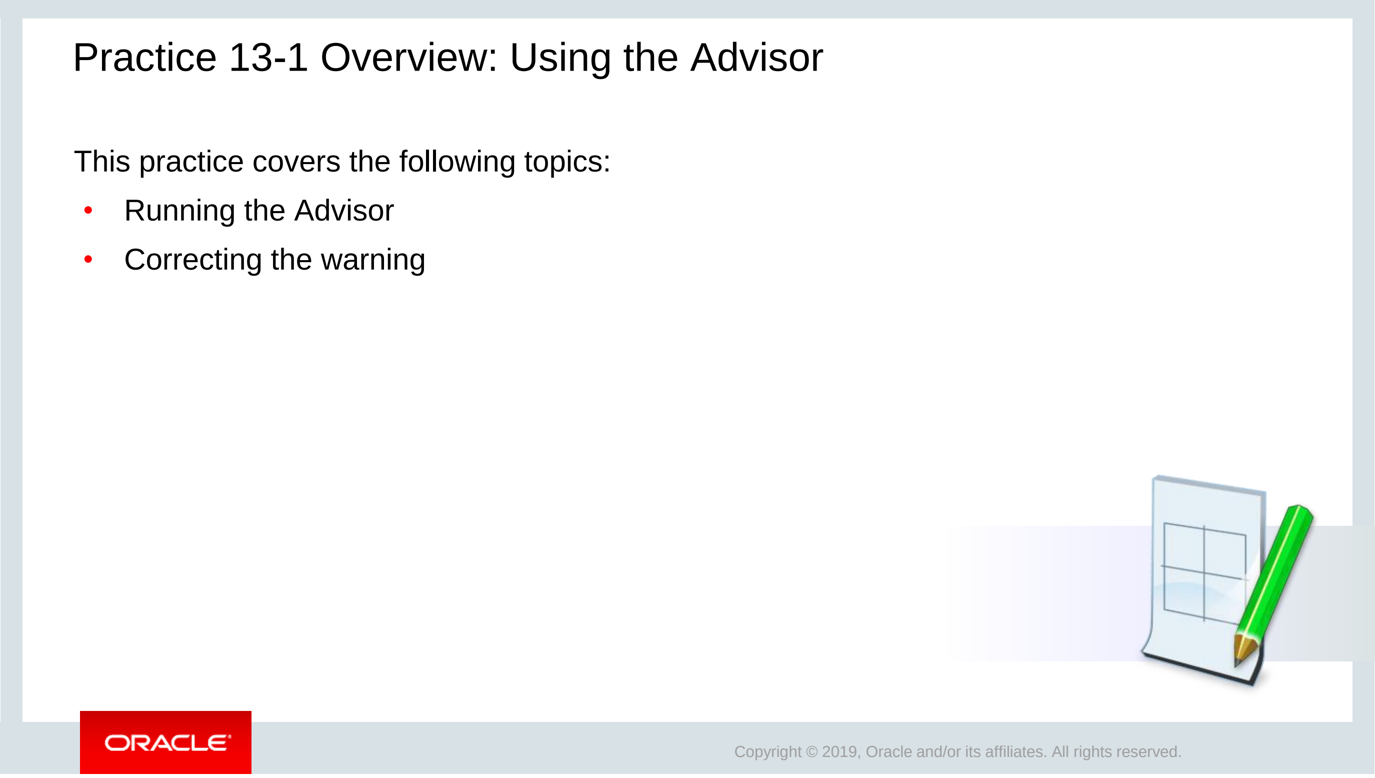
Note that after you run the Advisor on your application, your previous settings are recalled for the next use. If you do not want to perform a check on a particular violation, deselect the check box next to the violation. When there are no violations, you receive a message indicating that no errors or warnings were found.

Many of the checks are for informational purposes only and do not need to be resolved before deploying your application (unless you choose to do so).

Let's now see how Steve addresses the error message above:

1. In the *Advisor* window, click the **View** link. The *Page definition* for the page is displayed in the *Page Designer* view.
2. In the Rendering pane, *Page 8: Order Confirmation -> Order Lines* is selected by default in this example.
3. In the Property Editor pane, select **Source** -> **SQL Query** (code text box).
4. In the SQL Query (code text box), enter the following: `order by oi.order_id asc` (entering this SQL query will help sort the Orders by the Order ID and will also arrange them in an ascending order).
5. Save the page and rerun the Advisor to see that the violation is no longer in the list. You will see only the Authorization error, but as mentioned in the previous slide, we will discuss how to address this in a later lesson.

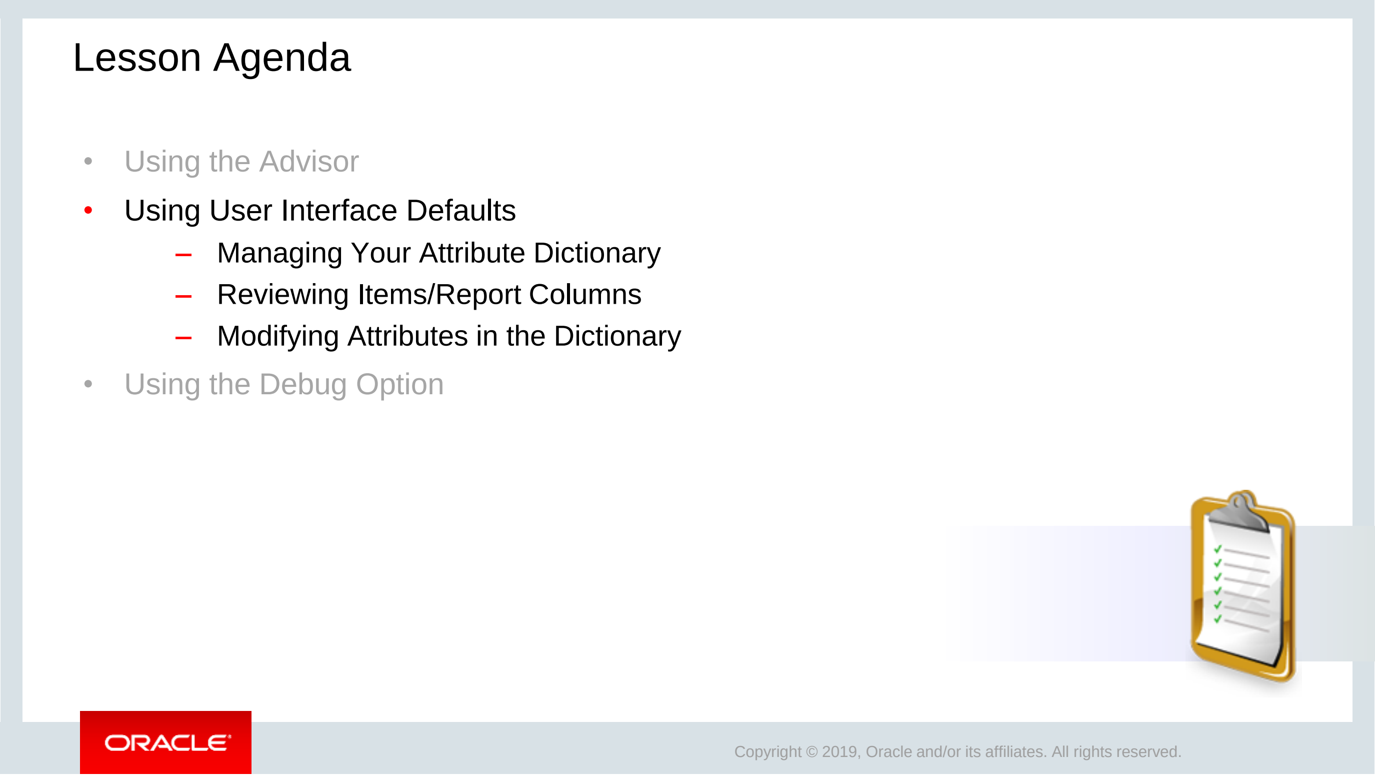
.



Practice 13-1 Overview: Using the Advisor

This practice covers the following topics:

- Running the Advisor
- Correcting the warning

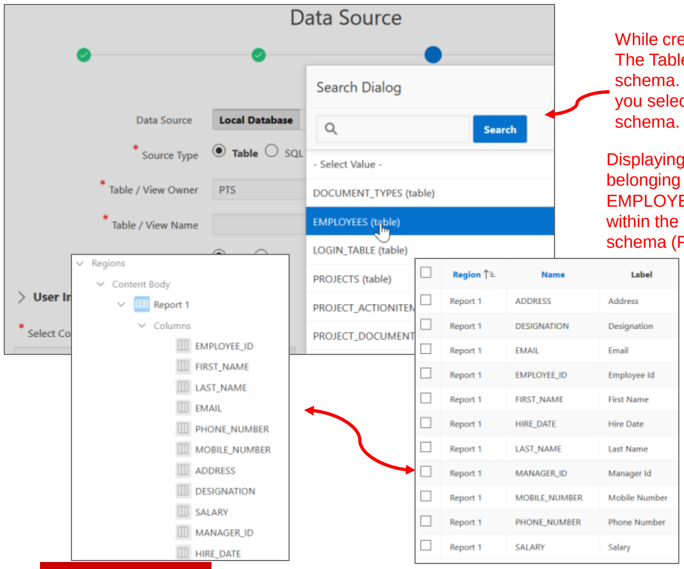


Lesson Agenda

- Using the Advisor
- Using User Interface Defaults
  - Managing Your Attribute Dictionary
  - Reviewing Items/Report Columns
  - Modifying Attributes in the Dictionary
- Using the Debug Option

Using User Interface Defaults

Enables you to assign default user interface properties to a table, column, or view within a specified schema. User Interface Defaults are divided into two categories - the Table Dictionary and the Attribute Dictionary.



Data Source

Data Source: Local Database

* Source Type: ● Table ○ SQL

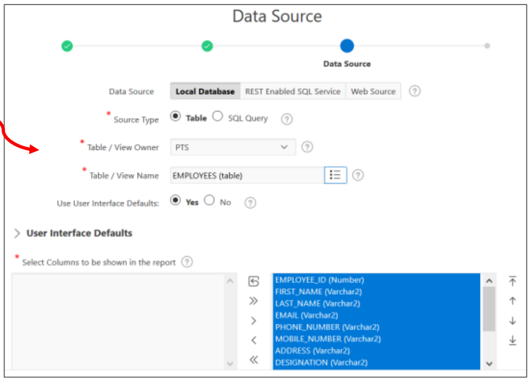
* Table / View Owner: PTS

* Table / View Name:

□	Region ↑	Name	Label
□	Report 1	ADDRESS	Address
□	Report 1	DESIGNATION	Designation
□	Report 1	EMAIL	Email
□	Report 1	EMPLOYEE_ID	Employee Id
□	Report 1	FIRST_NAME	First Name
□	Report 1	HIRE_DATE	Hire Date
□	Report 1	LAST_NAME	Last Name
□	Report 1	MANAGER_ID	Manager Id
□	Report 1	MOBILE_NUMBER	Mobile Number
□	Report 1	PHONE_NUMBER	Phone Number
□	Report 1	SALARY	Salary

While creating a Form or a Report, you select the Tables from the Table Dictionary. The Table Dictionary displays specific tables and columns within a selected schema. For example, if you want to create a report on the Employees database, you select the EMLOPYEES (table). The EMPLOYEES(table) belongs to the PTS schema.

Displaying columns belonging to the EMPLOYEES(table) within the selected schema (PTS).



Data Source

Data Source: Local Database | REST Enabled SQL Service | Web Source

* Source Type: ● Table ○ SQL Query

* Table / View Owner: PTS

* Table / View Name: EMPLOYEES (table)

Use User Interface Defaults: ● Yes ○ No

* Select Columns to be shown in the report

EMPLOYEE_ID (Number)
FIRST_NAME (Varchar2)
LAST_NAME (Varchar2)
EMAIL (Varchar2)
PHONE_NUMBER (Varchar2)
MOBILE_NUMBER (Varchar2)
ADDRESS (Varchar2)
DESIGNATION (Varchar2)

The Attribute Dictionary contains a set of attributes about a column that is used in creating forms and reports

You may have noticed that when you create a report or a form using a wizard in Application Express, the wizard uses the user interface defaults option to create default values for region and item properties. This saves your valuable development time, and it also has the added benefit of providing consistency across multiple pages in an application.

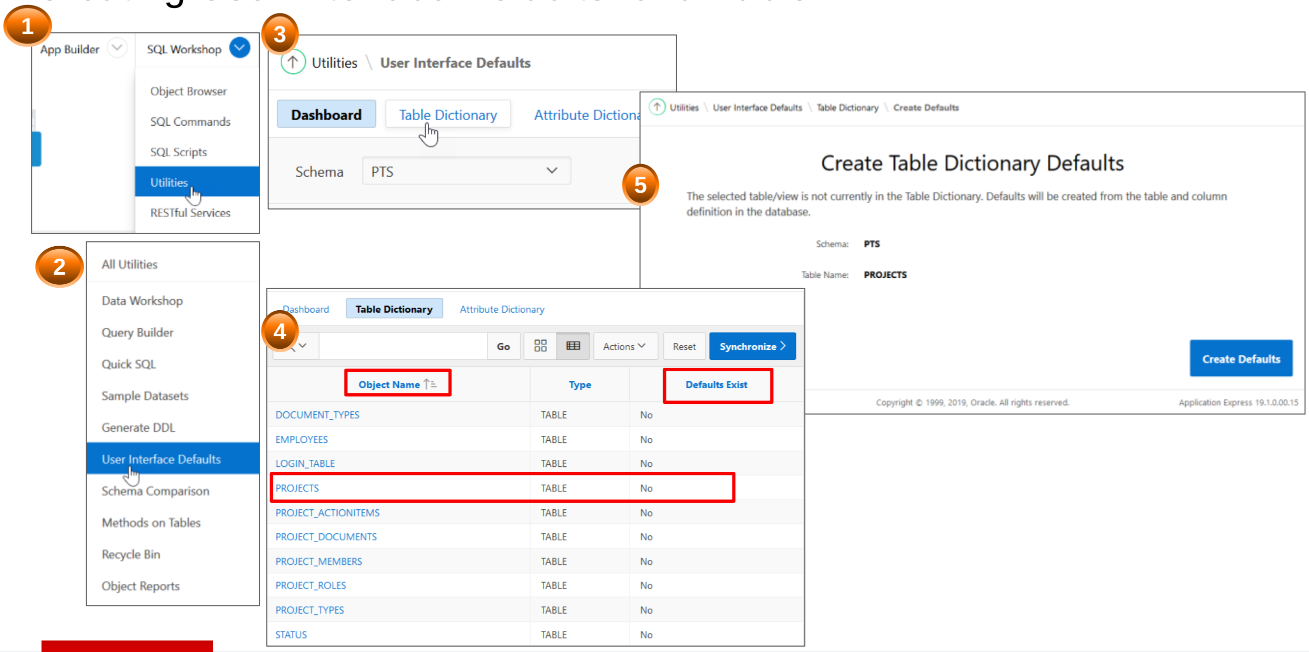
User Interface Defaults are divided into two categories: the Table Dictionary and the Attribute Dictionary.

- The *Table Dictionary* enables you to specify defaults for tables and columns within a selected schema. In the above example, you see `EMPLOYEES, PROJECTS, PROJECT_DOCUMENTS,` tables belonging to the PTS schema. When you select a particular table, say `EMPLOYEES,` the corresponding columns also get displayed as above. You can however select the columns that you would like to display in your report and form.
- The *Attribute Dictionary* contains a set of attributes about a column that you used while creating *Reports* and *Forms*. Some examples here are `EMAIL, FIRST_NAME, LAST_NAME.` The definitions are matched by column name, and a particular definition can be shared by several columns by using synonyms. You will learn how you can manage your attribute dictionary in the next couple of slides.

Note that the *Table Dictionary* takes priority over the *Attribute Dictionary* when you use user interface defaults while creating pages and regions.

In the next slide, you will learn how to create user interface defaults for a table.

Creating User Interface Defaults for a Table



The selected table/view is not currently in the Table Dictionary. Defaults will be created from the table and column definition in the database.

Schema: PTS

Table Name: PROJECTS

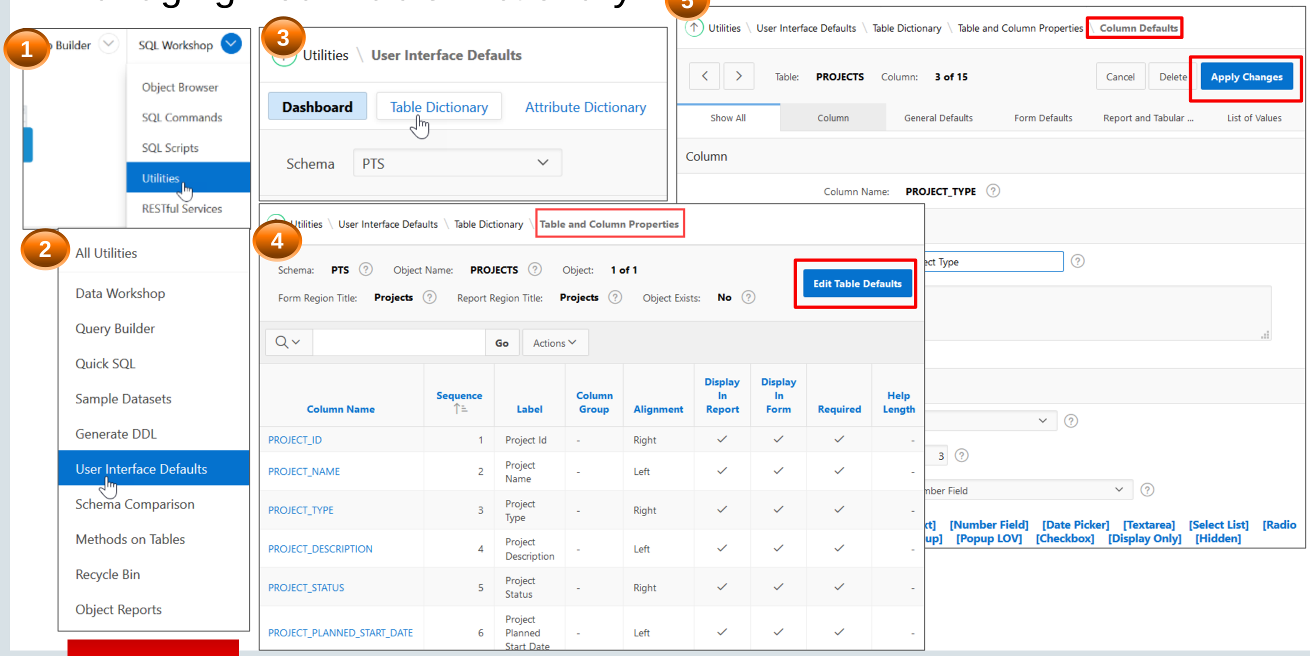
Object Name	Type	Defaults Exist
DOCUMENT_TYPES	TABLE	No
EMPLOYEES	TABLE	No
LOGIN_TABLE	TABLE	No
PROJECTS	TABLE	No
PROJECT_ACTIONITEMS	TABLE	No
PROJECT_DOCUMENTS	TABLE	No
PROJECT_MEMBERS	TABLE	No
PROJECT_ROLES	TABLE	No
PROJECT_TYPES	TABLE	No
STATUS	TABLE	No

If you have not created any user interface defaults for a table, you can use the Table Dictionary Create Defaults wizard to automatically generate defaults based on the database table and column definitions. To create the initial user interface defaults for tables:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **User Interface Defaults**. The Dashboard page for User Interface Defaults appears.
4. Click **Table Dictionary**. The Table Dictionary page displays a report of all objects in your workspace that includes a *Defaults Exist* column indicating whether defaults have been created for each object.
5. Click the object name for which you want to create defaults. In this example, Steve selects `PROJECTS` as no defaults exist for this object.
6. The Create Table Dictionary Defaults page appears. Click **Create Defaults**. The Table Dictionary page displays a report showing that defaults exist for the object you selected.

Note that after you create the initial defaults, you can modify the individual default values. In the next slide, you will learn how to do it.

Managing Your Table Dictionary



Schema: PTS

Table: PROJECTS Column: 3 of 15

Column Name: PROJECT_TYPE

Schema: PTS Object Name: PROJECTS Object: 1 of 1

Form Region Title: Projects Report Region Title: Projects Object Exists: No

Column Name	Sequence	Label	Column Group	Alignment	Display In Report	Display In Form	Required	Help Length
PROJECT_ID	1	Project Id	-	Right	✓	✓	✓	-
PROJECT_NAME	2	Project Name	-	Left	✓	✓	✓	-
PROJECT_TYPE	3	Project Type	-	Right	✓	✓	✓	-
PROJECT_DESCRIPTION	4	Project Description	-	Left	✓	✓	✓	-
PROJECT_STATUS	5	Project Status	-	Right	✓	✓	✓	-
PROJECT_PLANNED_START_DATE	6	Project Planned Start Date	-	Left	✓	✓	✓	-

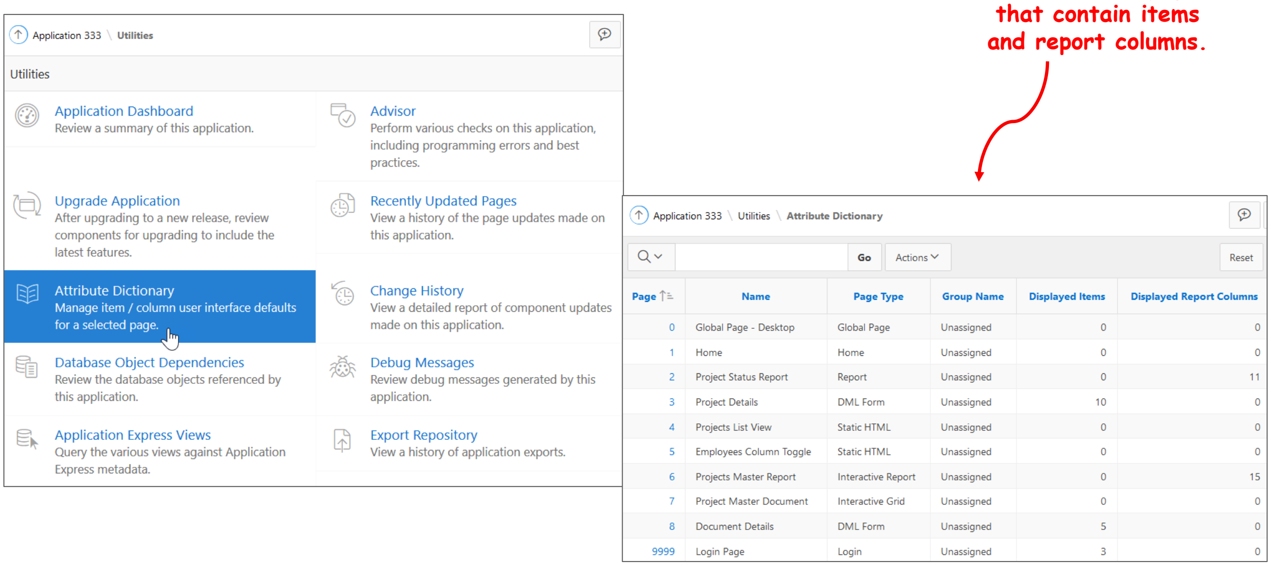
To modify existing table defaults in your Table Dictionary:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **User Interface Defaults**. The Dashboard page for User Interface Defaults appears.
4. Click **Table Dictionary**. The Table Dictionary page displays a report of all objects in your workspace. The Defaults Exist column indicates whether defaults have been created for an object.
5. Click the object name you want to modify. The **Table and Column Properties** page displays a report of column defaults.
6. Click the column name you want to modify.
7. The Column Defaults page is displayed. Make modifications and click **Apply Changes**.

Note that a column can be removed from the Table Dictionary, thus allowing the Attribute Dictionary defaults to be used during the creation process. For example, when you have auditing columns where you may want the exact same labels and help across every instance of CREATED_BY and CREATED_ON, regardless of which table they come from, you could simply remove the column defaults from each table for those columns.

In the next couple of slides, you will learn how to manage your Attribute Dictionary.

Managing Your Attribute Dictionary



View a list of pages that contain items and report columns.

Utilities

- Application Dashboard: Review a summary of this application.
- Advisor: Perform various checks on this application, including programming errors and best practices.
- Upgrade Application: After upgrading to a new release, review components for upgrading to include the latest features.
- Recently Updated Pages: View a history of the page updates made on this application.
- Attribute Dictionary: Manage item / column user interface defaults for a selected page.
- Change History: View a detailed report of component updates made on this application.
- Database Object Dependencies: Review the database objects referenced by this application.
- Debug Messages: Review debug messages generated by this application.
- Application Express Views: Query the various views against Application Express metadata.
- Export Repository: View a history of application exports.

Page	Name	Page Type	Group Name	Displayed Items	Displayed Report Columns
0	Global Page - Desktop	Global Page	Unassigned	0	0
1	Home	Home	Unassigned	0	0
2	Project Status Report	Report	Unassigned	0	11
3	Project Details	DML Form	Unassigned	10	0
4	Projects List View	Static HTML	Unassigned	0	0
5	Employees Column Toggle	Static HTML	Unassigned	0	0
6	Projects Master Report	Interactive Report	Unassigned	0	15
7	Project Master Document	Interactive Grid	Unassigned	0	0
8	Document Details	DML Form	Unassigned	5	0
9999	Login Page	Login	Unassigned	3	0

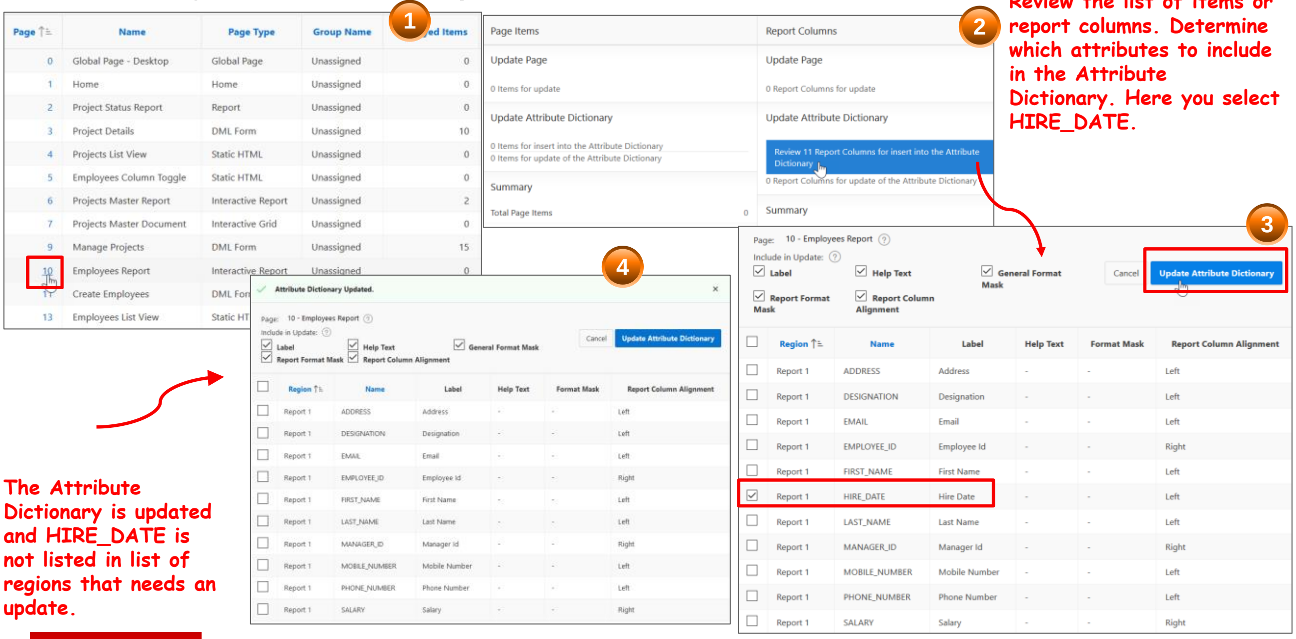
In the slide above, you see that the Attributes Dictionary page includes Page Items and Report Columns. You can use Page Item and Report Column definitions to update the Attribute Dictionary. You can also use the Attribute Dictionary to update Page Items and Report Columns. Hidden objects (that is, those in hidden regions and button items) are not counted in the number of candidate Items and Report Columns as these are not used in the updates.

To access the Attribute Dictionary of your application:

- Go to the application page.
- Select **Utilities**.
- Select **Attribute Dictionary**.

You can also access the Attribute Dictionary for a particular page by navigating to the page definition and selecting **Utilities** > **Attribute Dictionary**.

Reviewing Items and Report Columns



Page	Name	Page Type	Group Name	ed Items
0	Global Page - Desktop	Global Page	Unassigned	0
1	Home	Home	Unassigned	0
2	Project Status Report	Report	Unassigned	0
3	Project Details	DML Form	Unassigned	10
4	Projects List View	Static HTML	Unassigned	0
5	Employees Column Toggle	Static HTML	Unassigned	0
6	Projects Master Report	Interactive Report	Unassigned	2
7	Projects Master Document	Interactive Grid	Unassigned	0
9	Manage Projects	DML Form	Unassigned	15
10	Employees Report	Interactive Report	Unassigned	0
11	Create Employees	DML For		
13	Employees List View	Static HT		

Page Items	Report Columns
Update Page 0 Items for update	Update Page 0 Report Columns for update
Update Attribute Dictionary 0 Items for insert into the Attribute Dictionary 0 Items for update of the Attribute Dictionary	Update Attribute Dictionary Review 11 Report Columns for insert into the Attribute Dictionary 0 Report Columns for update of the Attribute Dictionary
Summary Total Page Items 0	Summary

Review the list of items or report columns. Determine which attributes to include in the Attribute Dictionary. Here you select HIRE_DATE.

Page: 10 - Employees Report

Include in Update:
☑ Label ☑ Help Text ☑ General Format Mask
☑ Report Format Mask ☑ Report Column Alignment

Cancel | Update Attribute Dictionary

□	Region	Name	Label	Help Text	Format Mask	Report Column Alignment
□	Report 1	ADDRESS	Address	-	-	Left
□	Report 1	DESIGNATION	Designation	-	-	Left
□	Report 1	EMAIL	Email	-	-	Left
□	Report 1	EMPLOYEE_ID	Employee Id	-	-	Right
□	Report 1	FIRST_NAME	First Name	-	-	Left
☑	Report 1	HIRE_DATE	Hire Date	-	-	Left
□	Report 1	LAST_NAME	Last Name	-	-	Left
□	Report 1	MANAGER_ID	Manager Id	-	-	Right
□	Report 1	MOBILE_NUMBER	Mobile Number	-	-	Left
□	Report 1	PHONE_NUMBER	Phone Number	-	-	Left
□	Report 1	SALARY	Salary	-	-	Right

Attribute Dictionary Updated.

Page: 10 - Employees Report

Include in Update:
☑ Label ☑ Help Text ☑ General Format Mask
☑ Report Format Mask ☑ Report Column Alignment

Cancel | Update Attribute Dictionary

□	Region	Name	Label	Help Text	Format Mask	Report Column Alignment
□	Report 1	ADDRESS	Address	-	-	Left
□	Report 1	DESIGNATION	Designation	-	-	Left
□	Report 1	EMAIL	Email	-	-	Left
□	Report 1	EMPLOYEE_ID	Employee Id	-	-	Right
□	Report 1	FIRST_NAME	First Name	-	-	Left
□	Report 1	LAST_NAME	Last Name	-	-	Left
□	Report 1	MANAGER_ID	Manager Id	-	-	Right
□	Report 1	MOBILE_NUMBER	Mobile Number	-	-	Left
□	Report 1	PHONE_NUMBER	Phone Number	-	-	Left
□	Report 1	SALARY	Salary	-	-	Right

The Attribute Dictionary is updated and HIRE_DATE is not listed in list of regions that needs an update.

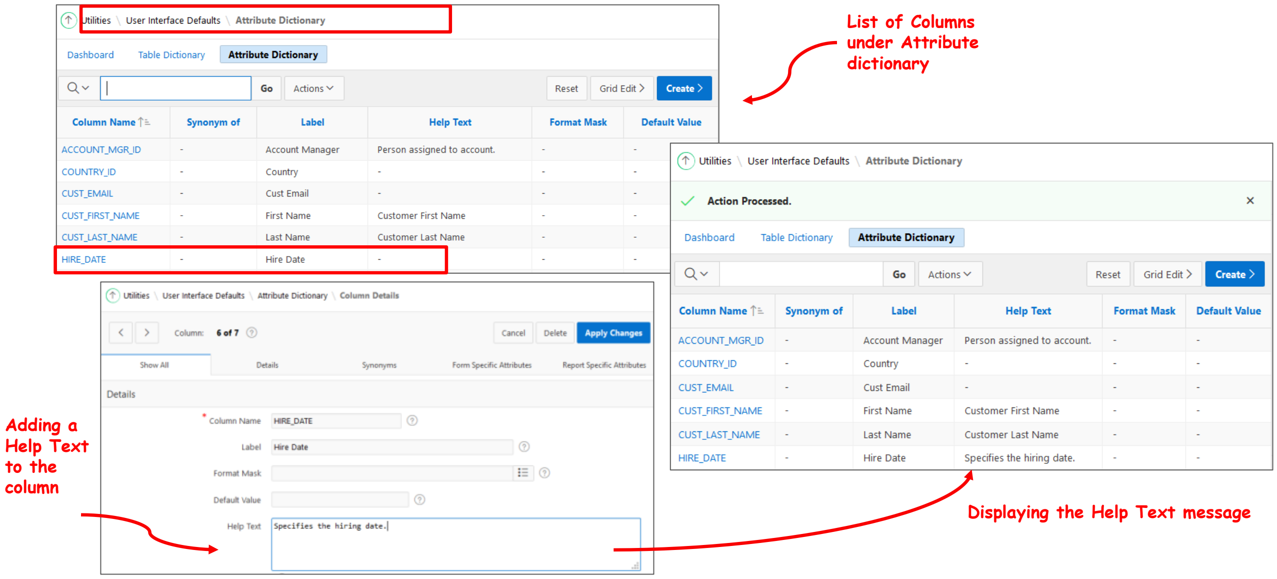
When you create an item on a page or create a report, you must review and evaluate which attributes you want to update in the dictionary. In the slide example, Steve updates the Attribute Dictionary for the `HIRE_DATE` item. He wants to include an update of all the attributes such as Label, Help text, General Format Mask, Report Format Mask, and Report Column Alignment for the `HIRE_DATE` item. Let's see how:

1. Select the check box for the appropriate row.
2. Click **Update Attribute Dictionary**. In this example, Steve selected `HIRE_DATE.` You get the status message as *Attribute Dictionary Updated*. Observe that the HIRE_DATE report region does not display anymore in the list of report regions that needs to be updated.

In the next slide, you will learn how to modify columns in the *Attribute Dictionary*.

Modifying Attributes in the Dictionary

You can modify attributes in the Attribute Dictionary using SQL Workshop > Utilities.



Utilities \ User Interface Defaults \ Attribute Dictionary
Dashboard
Table Dictionary
Attribute Dictionary
Go
Actions
Reset
Grid Edit
Create

Column Name	Synonym of	Label	Help Text	Format Mask	Default Value
ACCOUNT_MGR_ID	-	Account Manager	Person assigned to account.	-	-
COUNTRY_ID	-	Country	-	-	-
CUST_EMAIL	-	Cust Email	-	-	-
CUST_FIRST_NAME	-	First Name	Customer First Name	-	-
CUST_LAST_NAME	-	Last Name	Customer Last Name	-	-
HIRE_DATE	-	Hire Date	-	-	-

List of Columns under Attribute dictionary

Utilities \ User Interface Defaults \ Attribute Dictionary \ Column Details
Cancel
Delete
Apply Changes
Show All
Details
Synonyms
Form Specific Attributes
Report Specific Attributes
Details

Column: 6 of 7

* Column Name	HIRE_DATE
Label	Hire Date
Format Mask	
Default Value	
Help Text	Specifies the hiring date.

Adding a Help Text to the column

✓ Action Processed. ×

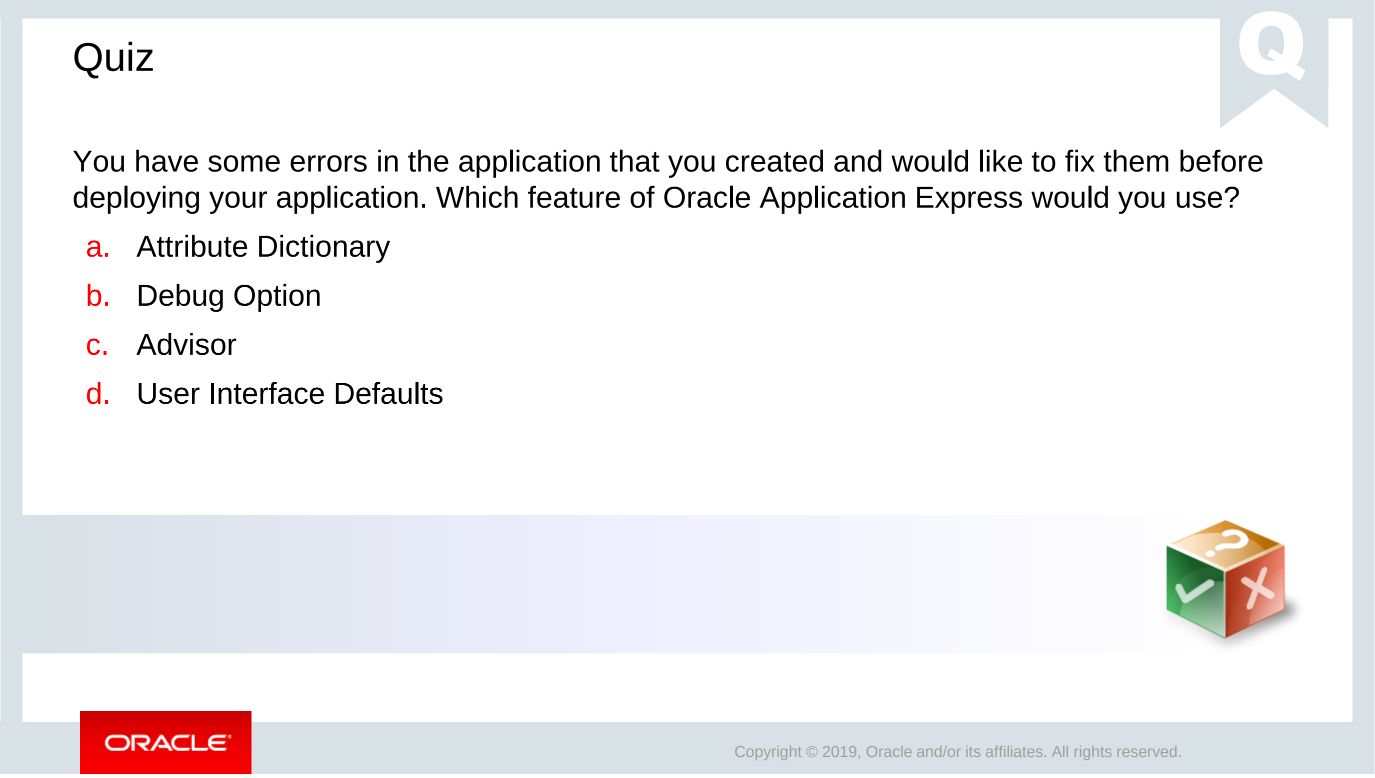
Column Name	Synonym of	Label	Help Text	Format Mask	Default Value
ACCOUNT_MGR_ID	-	Account Manager	Person assigned to account.	-	-
COUNTRY_ID	-	Country	-	-	-
CUST_EMAIL	-	Cust Email	-	-	-
CUST_FIRST_NAME	-	First Name	Customer First Name	-	-
CUST_LAST_NAME	-	Last Name	Customer Last Name	-	-
HIRE_DATE	-	Hire Date	Specifies the hiring date.	-	-

Displaying the Help Text message

What do you see in the above slide? Do you see that Steve here adds a Help Text for the `HIRE_DATE` item? This can be useful, for example, when you want to have a specific label or Help Text for the `HIRE_DATE` column in the `EMPLOYEES` table.

Let's see how to view a list of the columns in the *Attribute Dictionary* and modify it:

1. Select **SQL Workshop.**
2. Select **Utilities**.
3. Select **User Interface Defaults**.
4. Click the **Attribute Dictionary** tab.
5. To make changes to a column, click the link on the column name and make your changes (in this slide, Steve adds a Help Text to the `HIRE_DATE` column name).
6. Click **Apply Changes**.

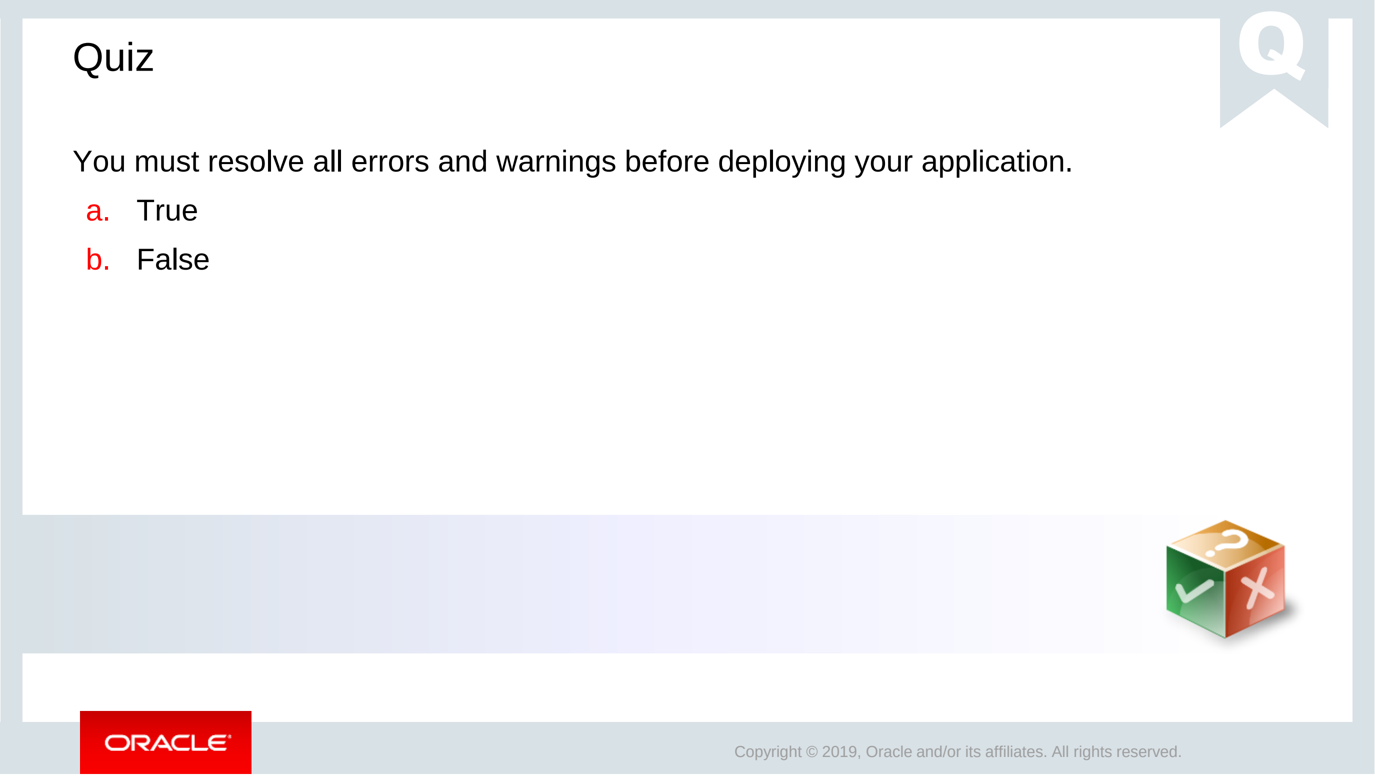


Quiz

You have some errors in the application that you created and would like to fix them before deploying your application. Which feature of Oracle Application Express would you use?

a. Attribute Dictionary
b. Debug Option
c. Advisor
d. User Interface Defaults

Answer: c



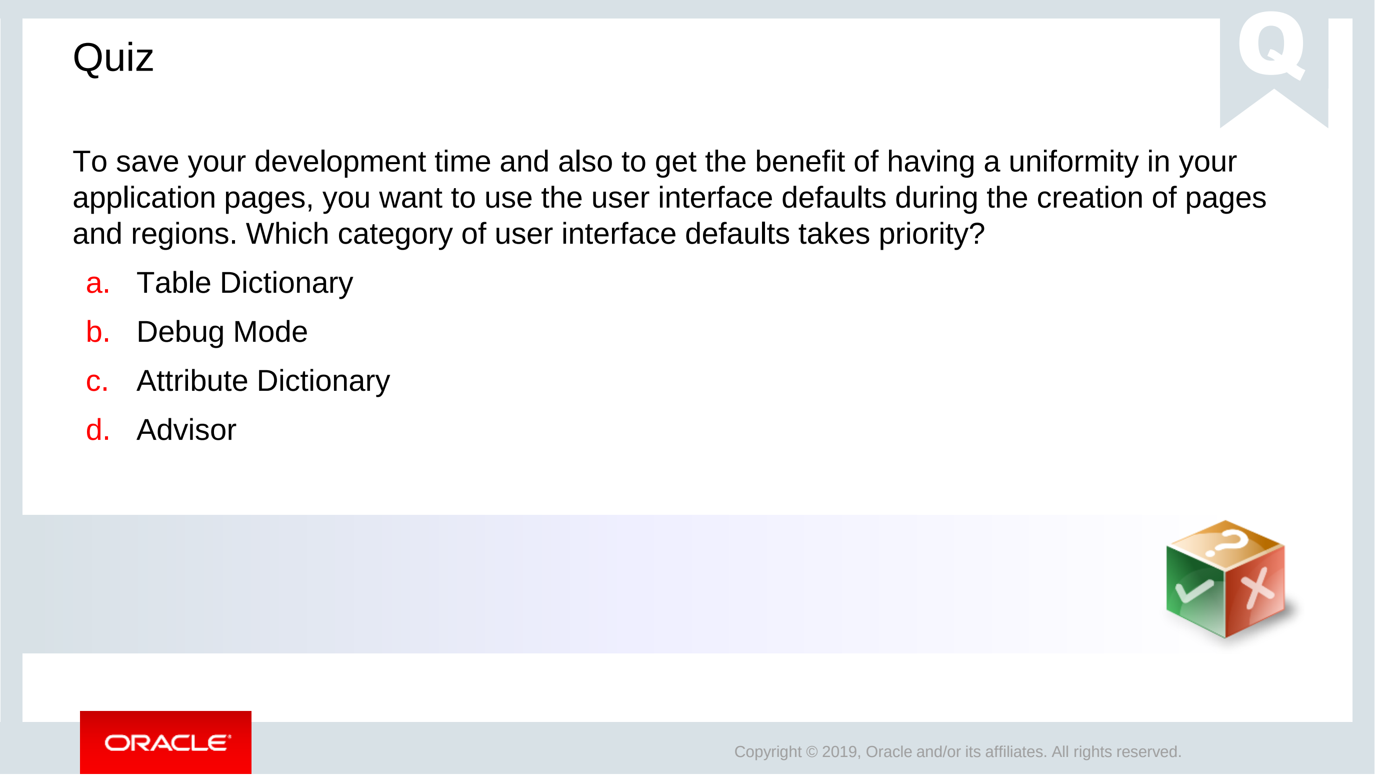
Quiz

You must resolve all errors and warnings before deploying your application.

a. True

b. False

Answer: b



Quiz

To save your development time and also to get the benefit of having a uniformity in your application pages, you want to use the user interface defaults during the creation of pages and regions. Which category of user interface defaults takes priority?

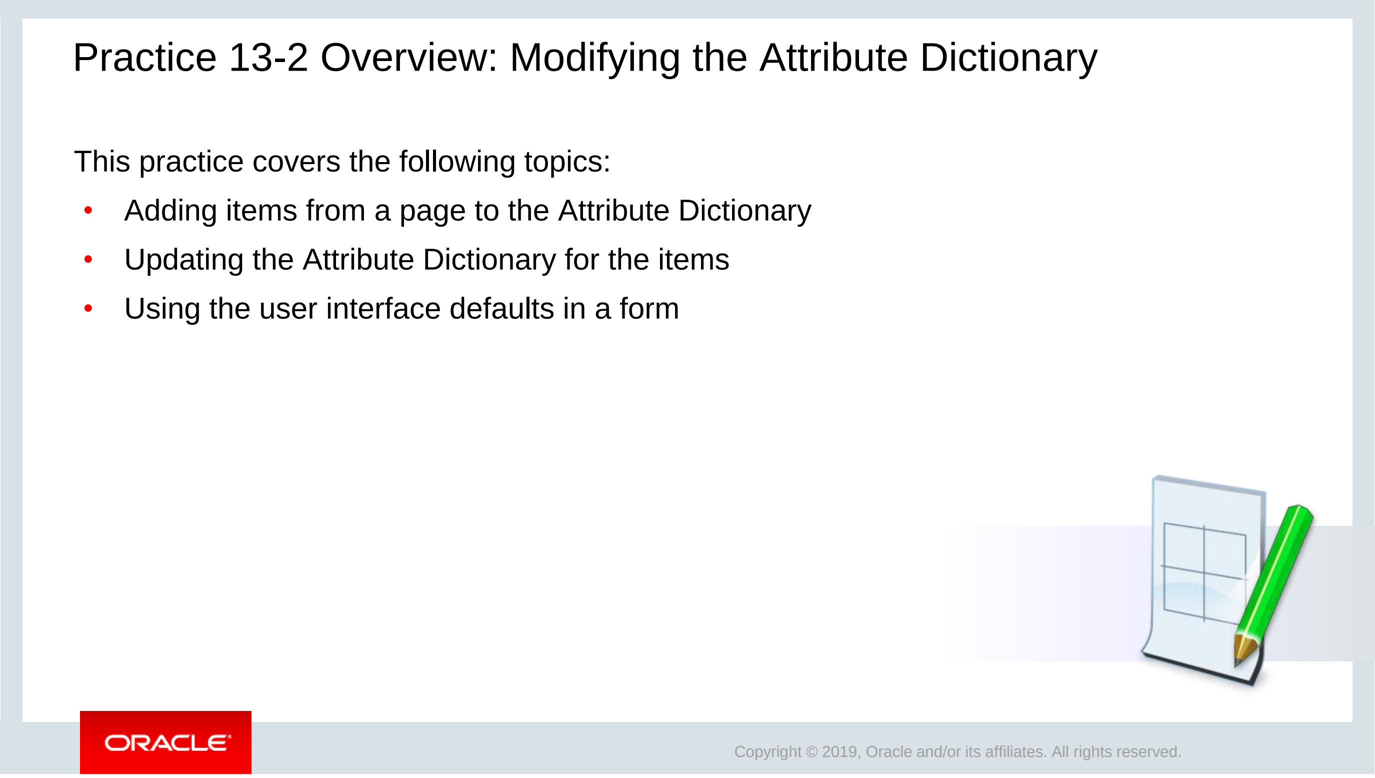
a. Table Dictionary

b. Debug Mode

c. Attribute Dictionary

d. Advisor

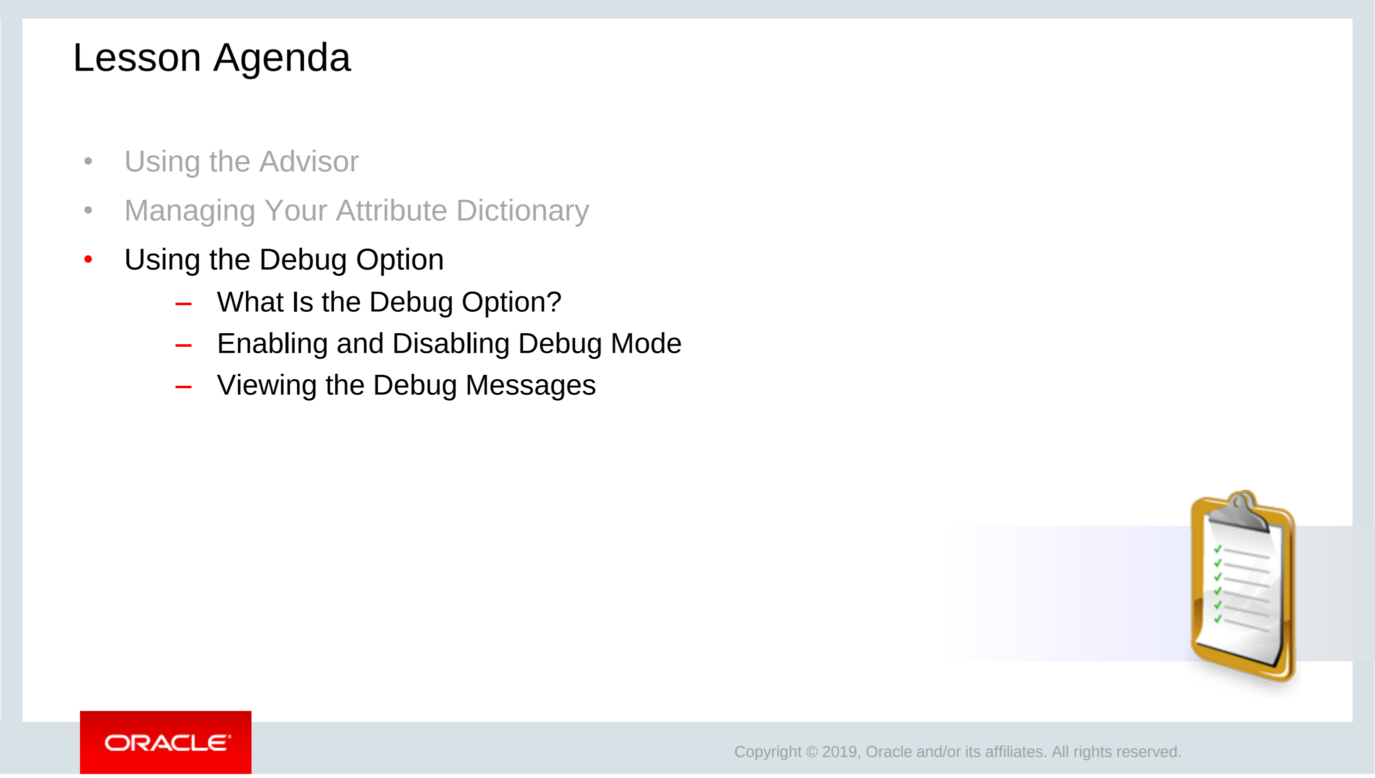
Answer: a



Practice 13-2 Overview: Modifying the Attribute Dictionary

This practice covers the following topics:

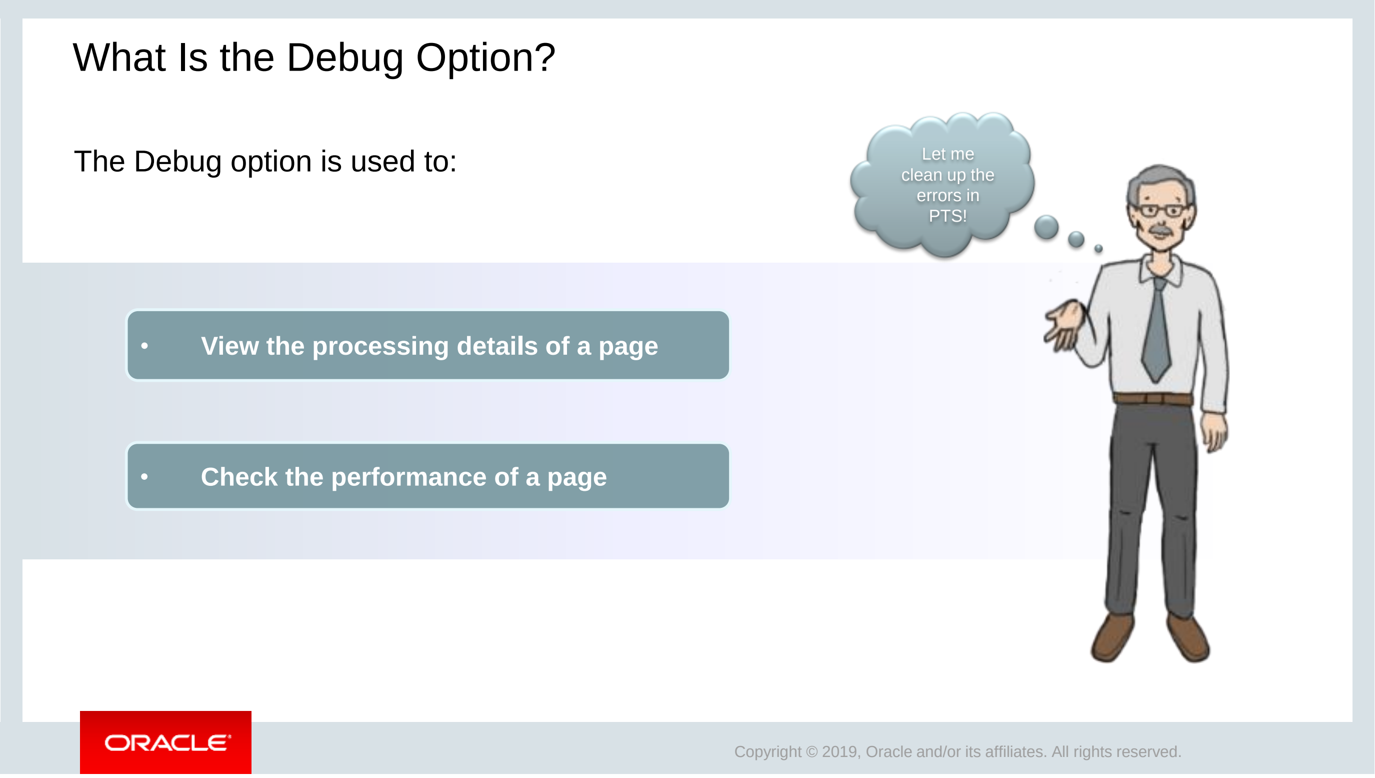
- Adding items from a page to the Attribute Dictionary
- Updating the Attribute Dictionary for the items
- Using the user interface defaults in a form



Lesson Agenda

- Using the Advisor
- Managing Your Attribute Dictionary
- Using the Debug Option
  - What Is the Debug Option?
  - Enabling and Disabling Debug Mode
  - Viewing the Debug Messages

ORACLE



What Is the Debug Option?

The Debug option is used to:

- View the processing details of a page
- Check the performance of a page

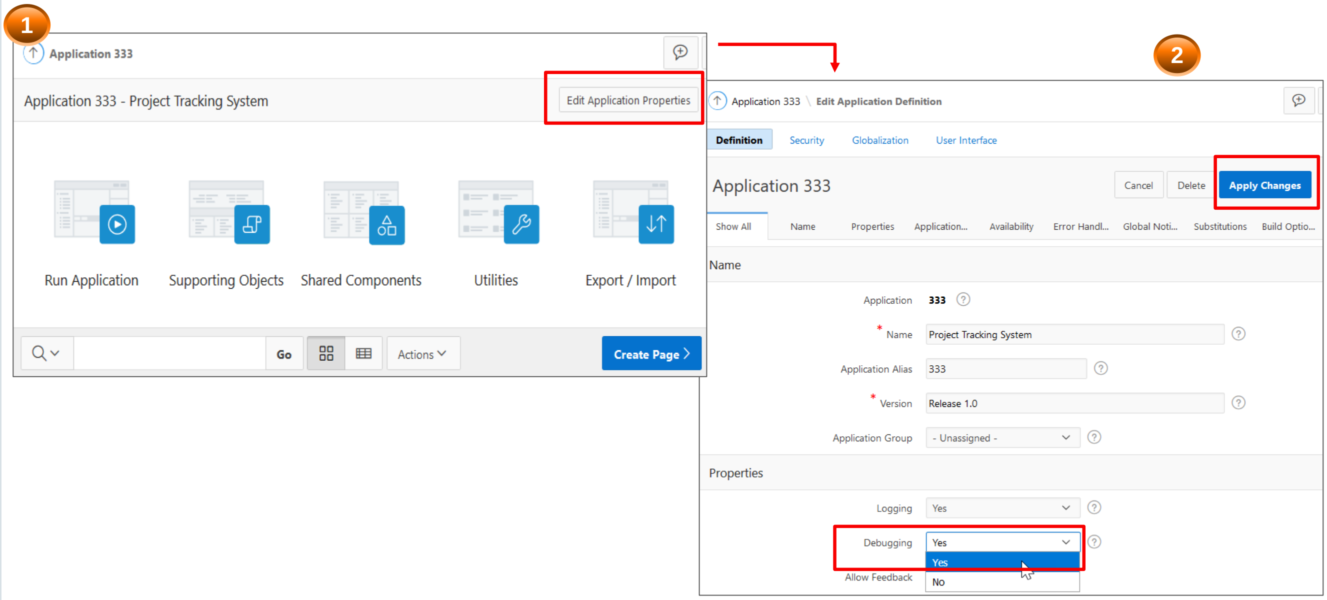
Let me clean up the errors in PTS!

Steve’s requirement was to view and debug his application before he deploys it. He can use the Debug mode’s built-in mechanism at run time to view the processing of a page and track down unexpected application behavior.

The Debug option provides useful information about what is happening in the background. In addition, you can use it to check the performance of a given page.

Let’s see how in the next few slides.

Enabling and Disabling Debug Mode



Application 333

* Name: Project Tracking System

Application Alias: 333

* Version: Release 1.0

Application Group: - Unassigned -

Logging: Yes

Debugging: Yes (Yes / No)

Running an application in *Debug* mode is useful when an application is under development. A developer who is logged in to the application's workspace can always run the application in debug mode.

You can configure whether end users can run the application in debug mode by using the Debugging attribute on the *Edit Application Properties* page. To enable or disable debugging feature in your application during development, perform the following steps:

1. On the application’s home page, click the **Edit Application Properties** button.
2. Click the Properties tab, select **Yes** for Debugging, and click **Apply Changes**.

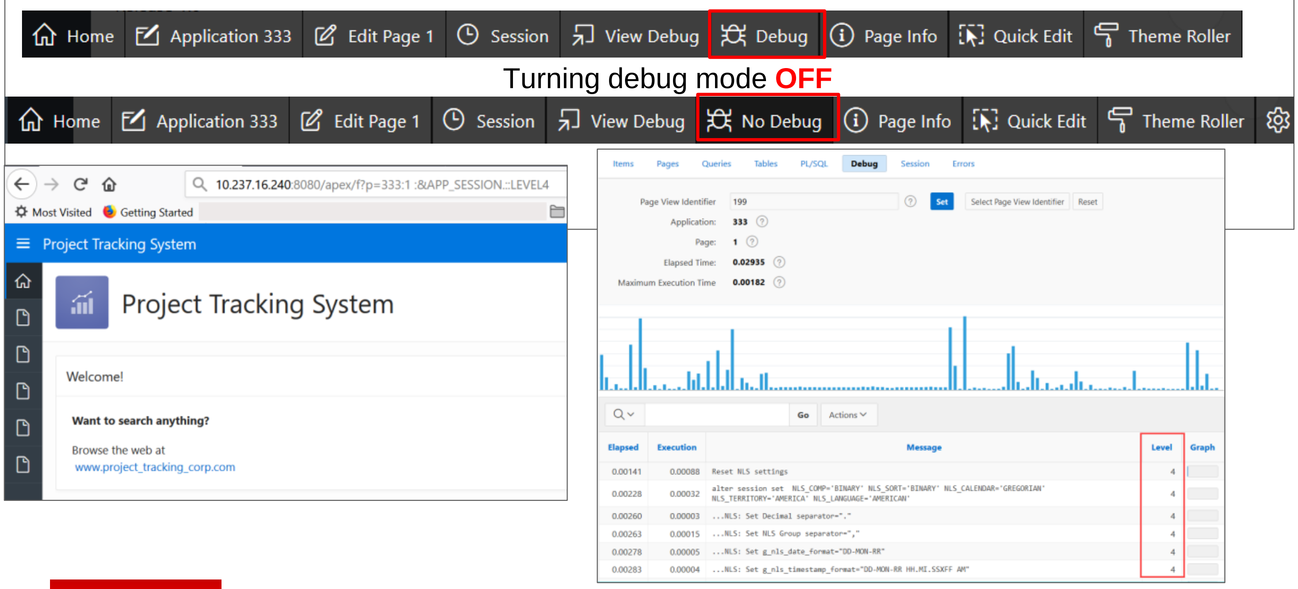
To disable the debugging option for an application, perform the same steps and set the debugging field to **No**.

Both end user and developer can debug an application. But it is the developer who decides if the end user can debug or not by setting the Application Properties.

Debugging an Application

Turning debug mode **ON**

Turning debug mode **OFF**



10.237.16.240:8080/apex/f?p=333:1 :&APP_SESSION.::LEVEL4

Page View Identifier: 199

Application: 333

Page: 1

Elapsed Time: 0.02935

Maximum Execution Time 0.00182

Elapsed	Execution	Message	Level	Graph
0.00141	0.00088	Reset NLS settings	4	
0.00228	0.00032	alter session set NLS_COMP='BINARY' NLS_SORT='BINARY' NLS_CALENDAR='GREGORIAN' NLS_TERRITORY='AMERICA' NLS_LANGUAGE='AMERICAN'	4	
0.00260	0.00003	...NLS: Set Decimal separator="."	4	
0.00263	0.00015	...NLS: Set NLS Group separator=","	4	
0.00278	0.00005	...NLS: Set g_nls_date_format="DD-MON-RR"	4	
0.00283	0.00004	...NLS: Set g_nls_timestamp_format="DD-MON-RR HH.MI.SSXFF AM"	4	

You can turn on and off the debug mode for an application at run time by:

- Clicking the **Debug** link on the Developer toolbar

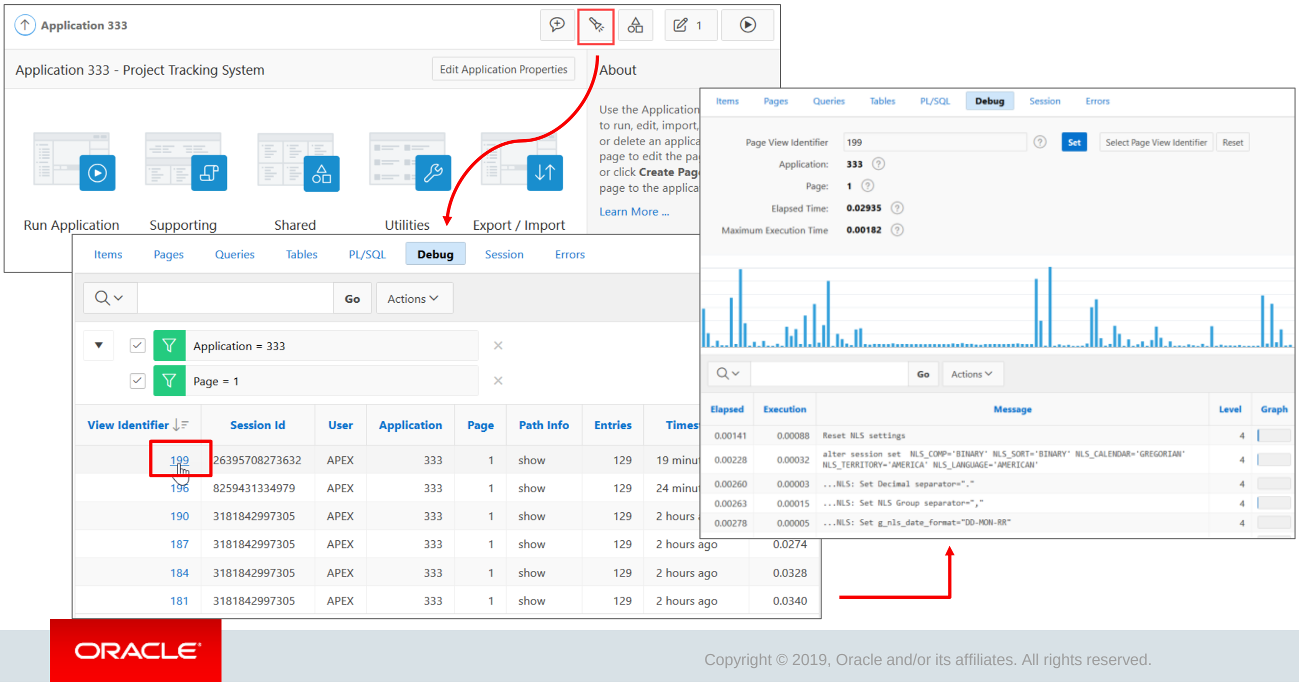
Note: To debug a page, you should click the Debug option before you make any changes in the page because the page will be reset when you do this. For example, if you make a change to a form, then click Debug. The page will be reset back to the original values in session state.

You can also use `f?p` syntax to run an application in Debugging mode. Simply call the page and set the Debug argument to **YES** or **LEVEL*n***, where `n` is between 1 (least detailed) and 9 (most detailed).

In this slide, Steve has set the Debug argument to `LEVEL4 (/f?p=333:1 :&APP_SESSION.::LEVEL4).` Therefore, when he clicks **View Debug** option on the Developer Toolbar, he gets to see all the relevant messages for that level.

Note that in the next couple of slides, you learn how to view Debug reports in Oracle Application Express.

Viewing Debug Reports



Application 333
Application 333 - Project Tracking System
Edit Application Properties
About
Use the Application to run, edit, import, or delete an applica page to edit the pa or click Create Pag page to the applica
Learn More ...
Run Application
Supporting
Shared
Utilities
Export / Import
Items
Pages
Queries
Tables
PL/SQL
Debug
Session
Errors
Go
Actions
Application = 333
Page = 1

View Identifier	Session Id	User	Application	Page	Path Info	Entries	Times	
199	26395708273632	APEX	333	1	show	129	19 minu	
196	8259431334979	APEX	333	1	show	129	24 minu	
190	3181842997305	APEX	333	1	show	129	2 hours	
187	3181842997305	APEX	333	1	show	129	2 hours ago	0.0274
184	3181842997305	APEX	333	1	show	129	2 hours ago	0.0328
181	3181842997305	APEX	333	1	show	129	2 hours ago	0.0340

Page View Identifier: 199
Application: 333
Page: 1
Elapsed Time: 0.02935
Maximum Execution Time 0.00182

Elapsed	Execution	Message	Level	Graph
0.00141	0.00088	Reset NLS settings	4	
0.00228	0.00032	alter session set NLS_COMP='BINARY' NLS_SORT='BINARY' NLS_CALENDAR='GREGORIAN' NLS_TERRITORY='AMERICA' NLS_LANGUAGE='AMERICAN'	4	
0.00260	0.00003	...NLS: Set Decimal separator="."	4	
0.00263	0.00015	...NLS: Set NLS Group separator=","	4	
0.00278	0.00005	...NLS: Set g_nls_date_format="DD-MON-RR"	4	

Let's now see how Steve views his *Debug* reports. He can view the reports either/or from the:

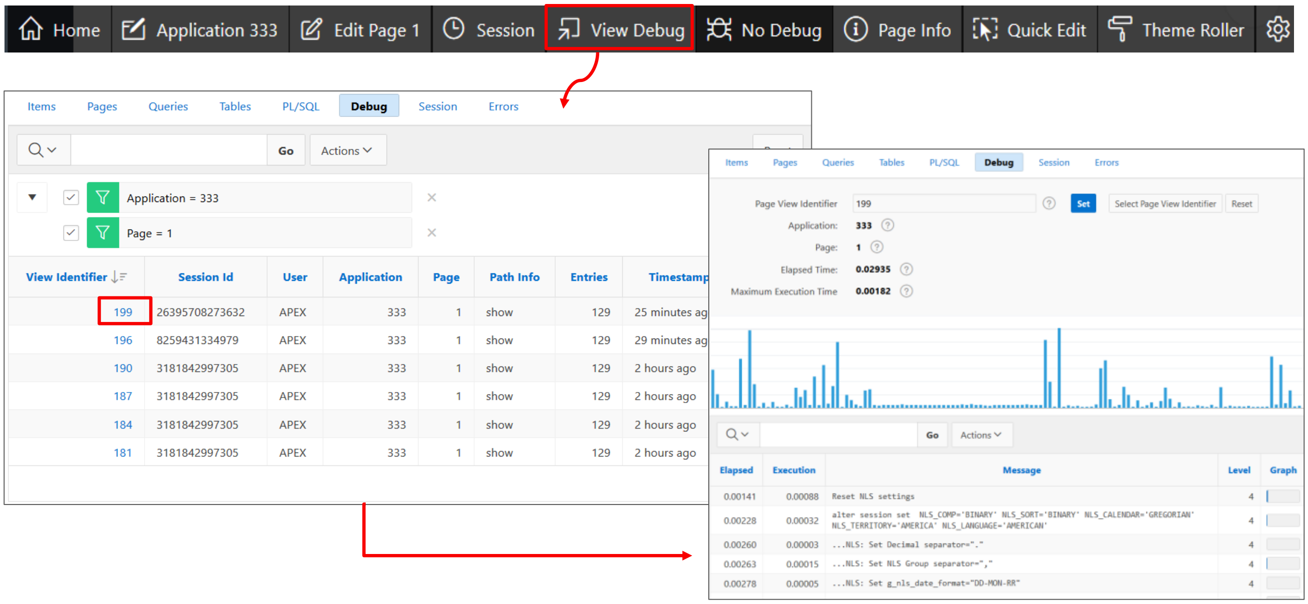
- Application Page
- Runtime Developer Toolbar

If an application runs in debug mode, Oracle Application Express writes information about how it processes each page request to a log table. To view debug reports:

1. Click the **Find** icon.
2. The **Items Finder** appears.
3. Click the **Debug** tab.
4. Click the **View Identifier** to the left of the session you want to view. The Debug report appears.
5. You can place the cursor over the graph to view additional details.
6. To go to that step in the report, click the bar. Note that debug sessions listed in the report live for at least 2 weeks and age out along with the activity log. This is usually more than adequate for debugging purposes.

In the next slide, you learn how to view the Debug reports from a runtime environment.

Viewing Debug Reports (at Runtime)



Application = 333

Page = 1

View Identifier	Session Id	User	Application	Page	Path Info	Entries	Timestamp
199	26395708273632	APEX	333	1	show	129	25 minutes ag
196	8259431334979	APEX	333	1	show	129	29 minutes ag
190	3181842997305	APEX	333	1	show	129	2 hours ago
187	3181842997305	APEX	333	1	show	129	2 hours ago
184	3181842997305	APEX	333	1	show	129	2 hours ago
181	3181842997305	APEX	333	1	show	129	2 hours ago

Page View Identifier 199

Application: 333

Page: 1

Elapsed Time: 0.02935

Maximum Execution Time 0.00182

Elapsed	Execution	Message	Level	Graph
0.00141	0.00088	Reset NLS settings	4	
0.00228	0.00032	alter session set NLS_COMP='BINARY' NLS_SORT='BINARY' NLS_CALENDAR='GREGORIAN' NLS_TERRITORY='AMERICA' NLS_LANGUAGE='AMERICAN'	4	
0.00260	0.00003	...NLS: Set Decimal separator="."	4	
0.00263	0.00015	...NLS: Set NLS Group separator=","	4	
0.00278	0.00005	...NLS: Set g_nls_date_format="DD-MON-RR"	4	

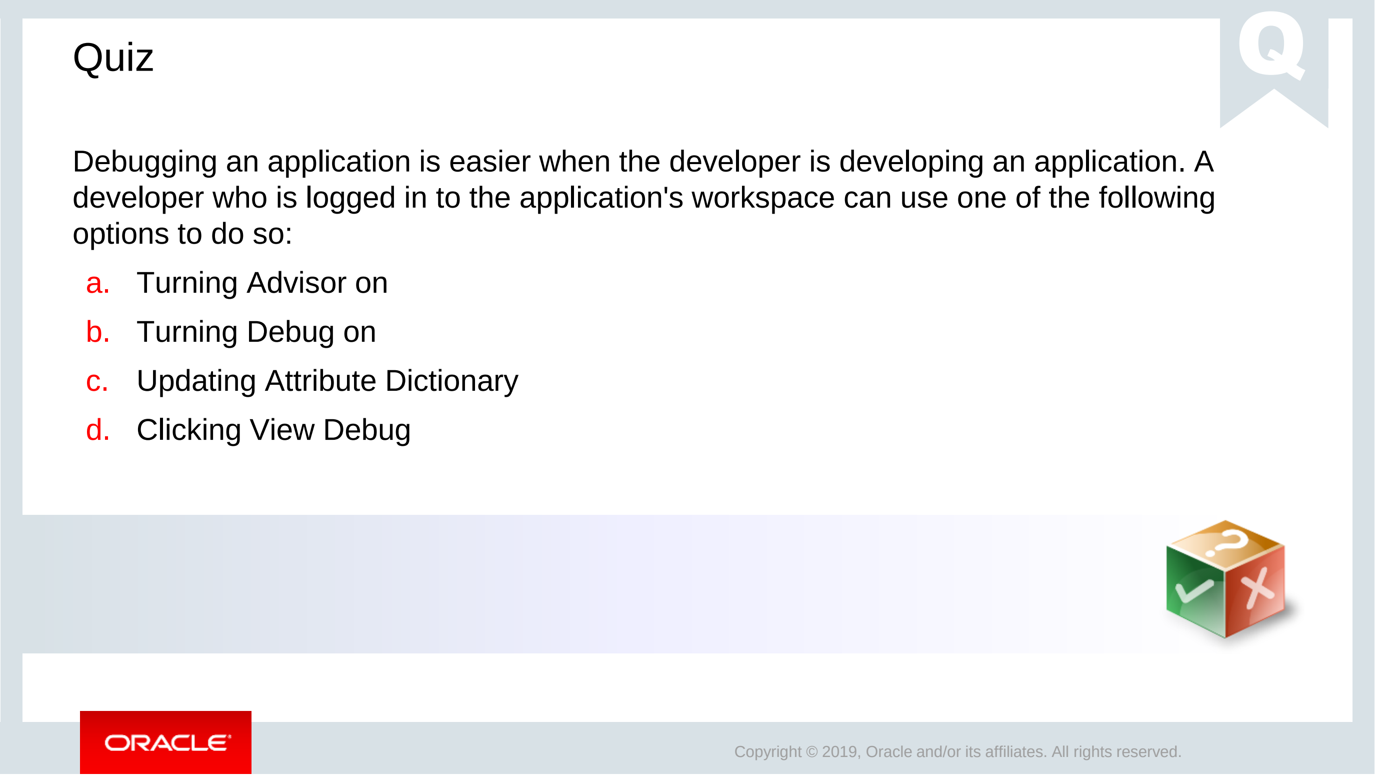
To view debug messages from a runtime environment:

1. Locate and run the application in Debug mode.
2. Click **View Debug** on the Runtime Developer toolbar. A list of Debug messages are shown.
3. Click the **View Identifier** to the left of the session you want to view. The Debug Message Data page appears.
4. You can place the cursor over the graph to view additional details.
5. To go to that step in the report, click the bar.

When a page is rendered (in this example, Page 1), a set of messages is displayed, which includes the following:

- NLS Language messages
- Authentication messages
- Session state messages
- `BEFORE_HEADER` and `AFTER_HEADER` processing messages for any branching, computations, and processes
- Region
- Item
- `BEFORE_FOOTER` and `AFTER_FOOTER` processing messages for any branching, computations, and processes

In addition to the preceding messages, the execution timing per second is displayed to make it clear how long each process is taking.



Quiz

Debugging an application is easier when the developer is developing an application. A developer who is logged in to the application's workspace can use one of the following options to do so:

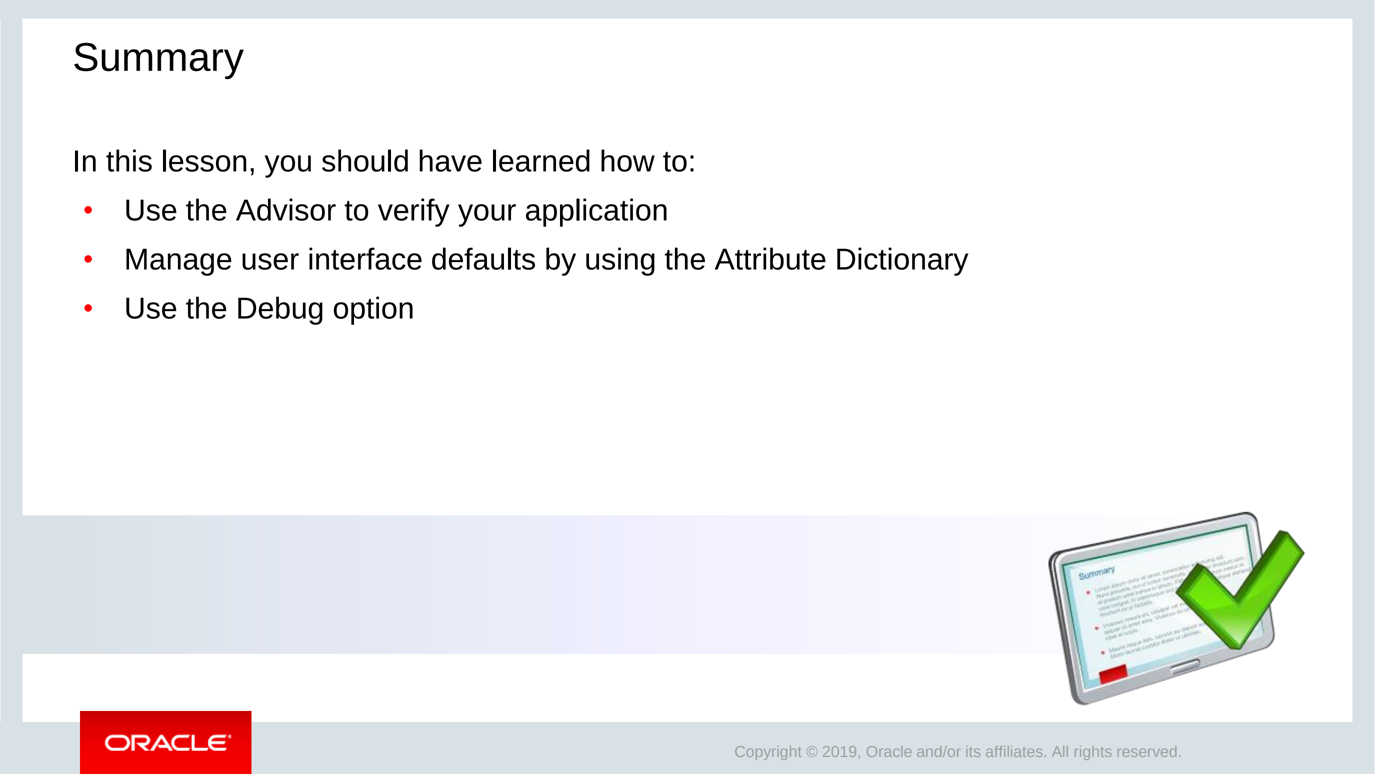
a. Turning Advisor on

b. Turning Debug on

c. Updating Attribute Dictionary

d. Clicking View Debug

Answer: b



Summary

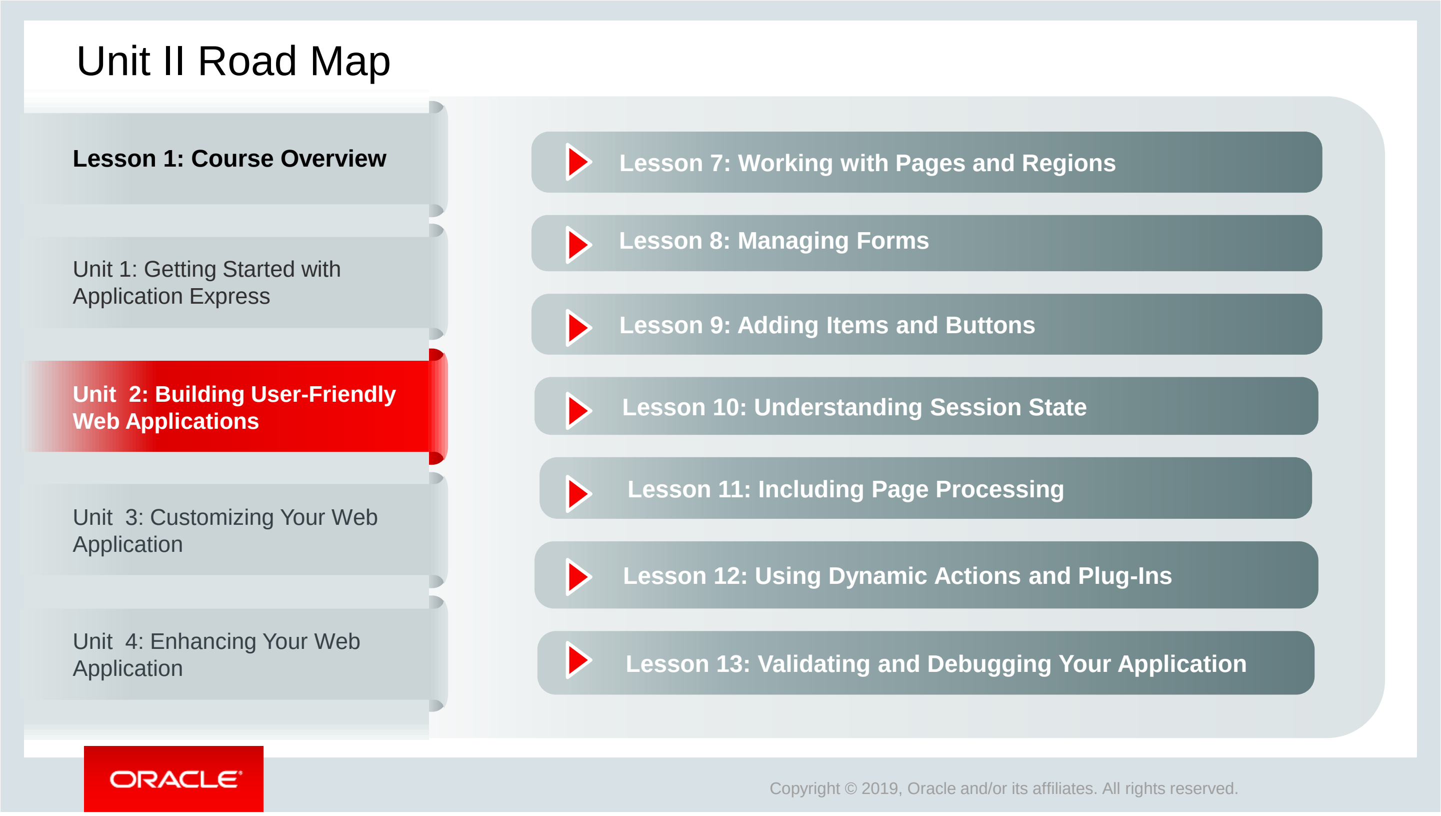
In this lesson, you should have learned how to:

- Use the Advisor to verify your application
- Manage user interface defaults by using the Attribute Dictionary
- Use the Debug option

In this lesson, you learned about using the Advisor, managing user interface defaults using the Attribute Dictionary, and also using the Debug option.

II

Unit II Summary: Building User-Friendly Web Applications



Unit II Road Map

Lesson 1: Course Overview

Unit 1: Getting Started with Application Express

Unit 2: Building User-Friendly Web Applications

Unit 3: Customizing Your Web Application

Unit 4: Enhancing Your Web Application

- **Lesson 7: Working with Pages and Regions**
- **Lesson 8: Managing Forms**
- **Lesson 9: Adding Items and Buttons**
- **Lesson 10: Understanding Session State**
- **Lesson 11: Including Page Processing**
- **Lesson 12: Using Dynamic Actions and Plug-Ins**
- **Lesson 13: Validating and Debugging Your Application**

In Unit 2, you completed seven topics.