

Practice: Getting Started with Oracle APEX on the Oracle Autonomous Database

Practice 1: Provision an APEX Workspace

Introduction

Oracle APEX is a low-code application platform for Oracle Database. APEX Application Development, Autonomous Data Warehouse (ADW), and Autonomous Transaction Processing (ATP) are fully managed services, pre-integrated and pre-configured with APEX, for rapidly building and deploying modern data-driven applications in Oracle Cloud. Business users, citizens, and application developers can create enterprise apps 20X faster with 100X less code — without having to learn complex web technologies with just a browser. To start, you will need to decide on the service you are going to use for this workshop and then create an APEX workspace accordingly.

What is an APEX Workspace?

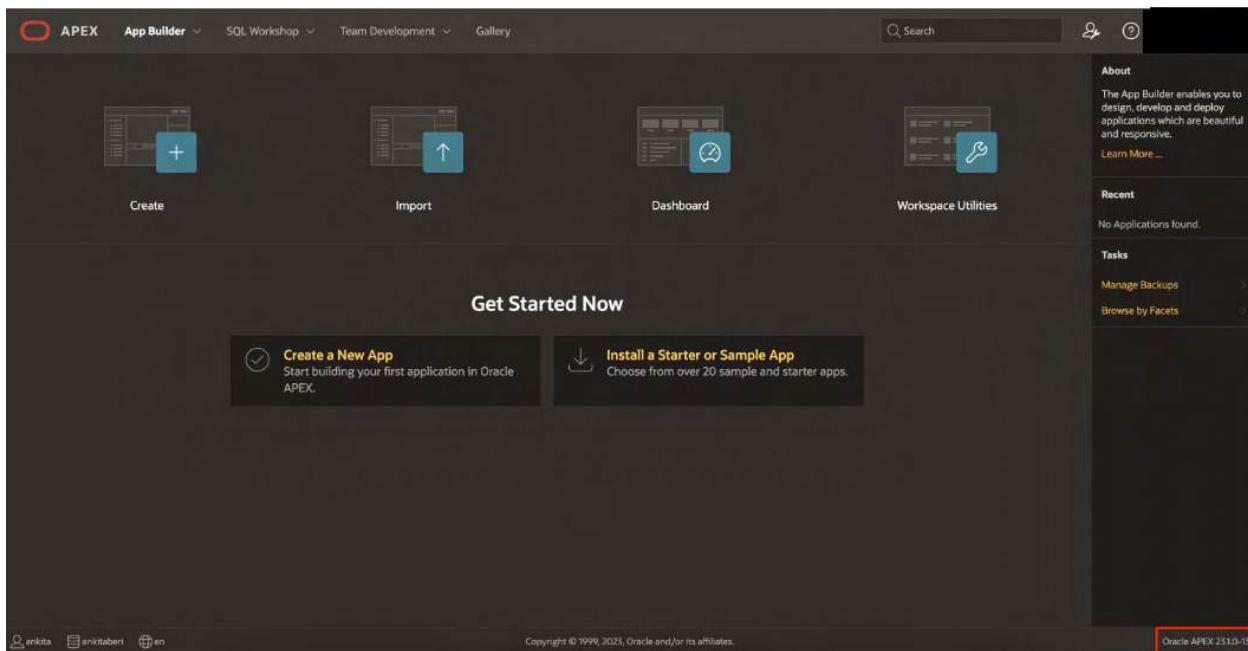
An APEX Workspace is a logical domain where you define APEX applications. Each workspace is associated with one or more database schemas (database users) which are used to store the database objects, such as tables, views, packages, and more. APEX applications are built on top of these database objects.

How do I find my APEX release version?

To determine which release of Oracle APEX you are currently running, do one of the following:

- **View the release number on the Workspace home page:**

- Sign in to Oracle APEX. The Workspace home page appears. The current release version is displayed in the bottom right corner.



Oracle APEX 23.1.0-13

- **View the About APEX page:**

- Sign in to Oracle APEX. The Workspace home page appears.
- Click the Help menu at the top of the page and select About. The About APEX page appears.

The screenshot shows the 'About Oracle APEX' page with a dark background. At the top is a logo consisting of a yellow pencil and a red X. Below it is the text 'Oracle APEX'. A table titled 'Details' follows, listing the following information:

| Details | |
|----------------------|------------------------|
| Product Build | 23.1.0-13 |
| Schema Compatibility | 2023.04.28 |
| Last DDL Time | 04/13/2023 02:40:47 PM |
| Host Schema | APEX_PUBLIC_USER |
| Application Owner | APEX_230100 |
| Workspace ID | 3418696304433886 |
| Workspace Name | ANKITABERI |

Where to Run the Lab

You can run this lab in any Oracle Database with APEX 23.1 installed. This includes the APEX Application Development Service, the free, "Development Only" apex.oracle.com service, your on-premises Oracle Database (providing APEX 23.1 is installed), on a third-party cloud provider where APEX 23.1 is installed, or even on your laptop by installing Oracle XE or Oracle VirtualBox App Dev VM and installing APEX 23.1.

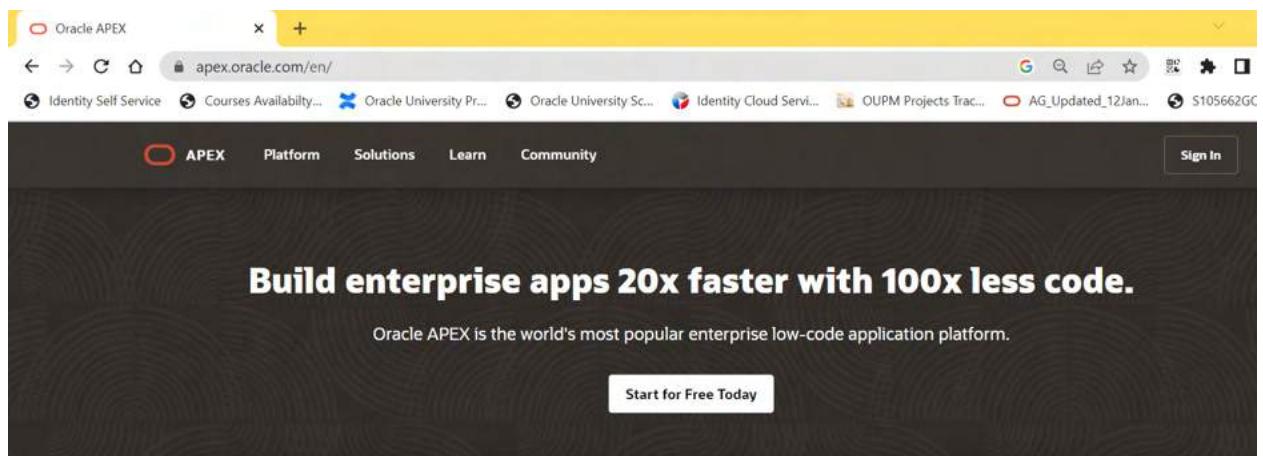
For this course, we will use the free, "Development Only" apex.oracle.com service.

Sign Up for apex.oracle.com

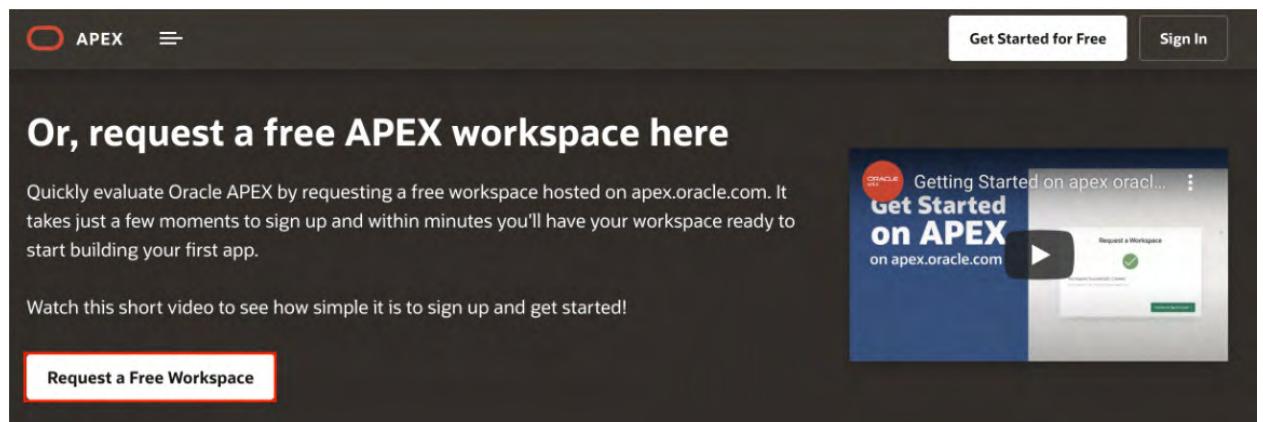
Signing up for apex.oracle.com is simply a matter of providing details on the workspace you wish to create and then waiting for the approval email.

1. Go to <https://apex.oracle.com>

2. Click **Get Started for Free**.



3. Scroll down and you will see details for apex.oracle.com. Click **Request a Free Workspace**.

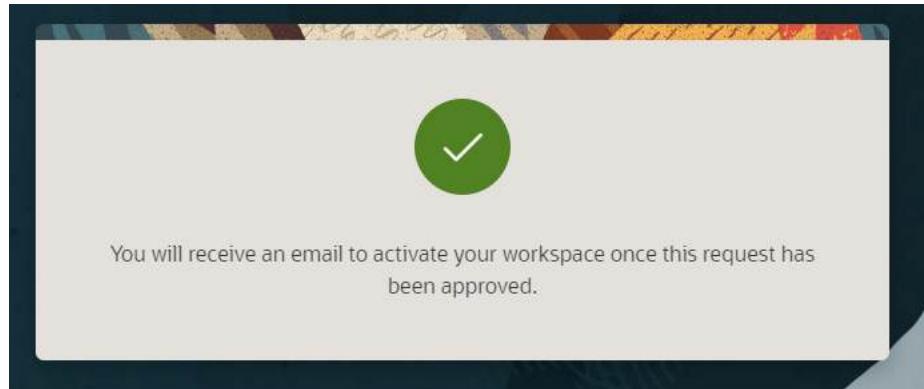


4. On the Request a Workspace dialog, enter your identification details – First Name, Last Name, Email, Workspace Name, Your Location, etc.

Now, please accept the **Terms** of the **Oracle APEX Service Agreement** and then **Request Workspace**

Note: For workspace, enter a unique name, such as first initial and last name.

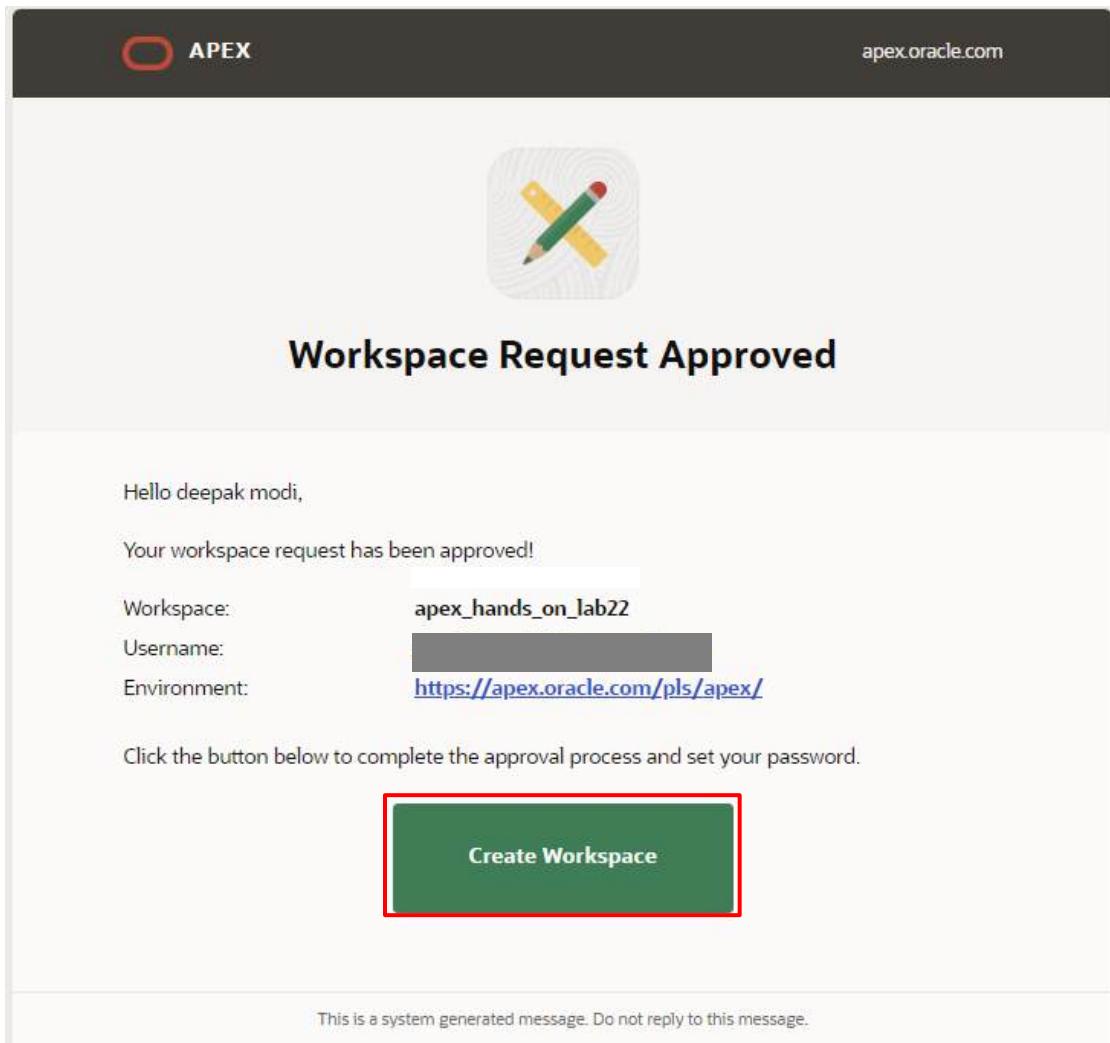
The screenshot shows the Oracle APEX 'Request a workspace' dialog. The dialog has several input fields: 'First name' and 'Last name' (both highlighted with red boxes), 'Email' (also highlighted with a red box), 'Workspace name' (containing 'apex_handson_labs'), 'Your Location' (set to 'India'), and two radio buttons for 'Are you new to Oracle APEX?' ('Yes' is selected). Below these are two questions with radio button options: 'Do you plan to use APEX for education or training?' ('Yes' is selected) and 'I agree to the terms of the Oracle APEX Service Agreement' (checkbox is checked). At the bottom right is a 'Request Workspace' button.



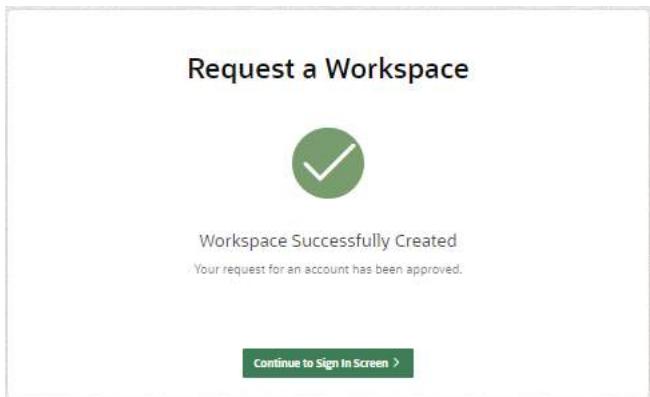
5. Check your email. You should get an email from apexext-noreply@ociapp-notifications.oracle.com within a few minutes.

Note: If you don't get an email go back to Step 3 and make sure to enter your email correctly.

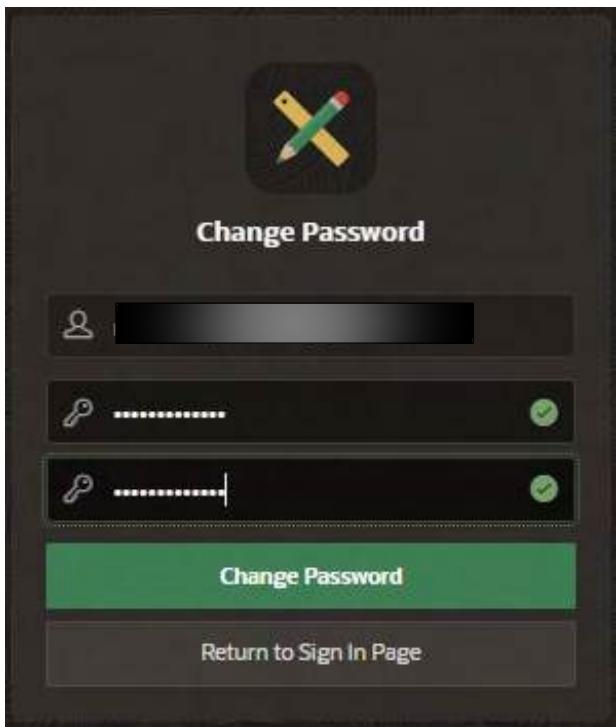
Within the email body, click **Create Workspace**.



6. Click **Continue to Sign In Screen**.



7. Enter your password, and click **Change Password**.



- On the APEX Workspace login page, enter your password, select the **Remember workspace and username** checkbox, and then click Sign In.



- You should now be in the APEX Builder.

Summary

At this point, you know how to create an APEX Workspace and you are ready to start building amazing apps, fast. You may now **proceed to the next practice**.

Practice 2: Install and Run a Sample Application Overview

In this lab, you will navigate through the major components of Oracle APEX. This lab also covers installing and running a sample application.

In this lab, you will:

- Navigate through the major Components of Oracle APEX
- Install and run a packaged application

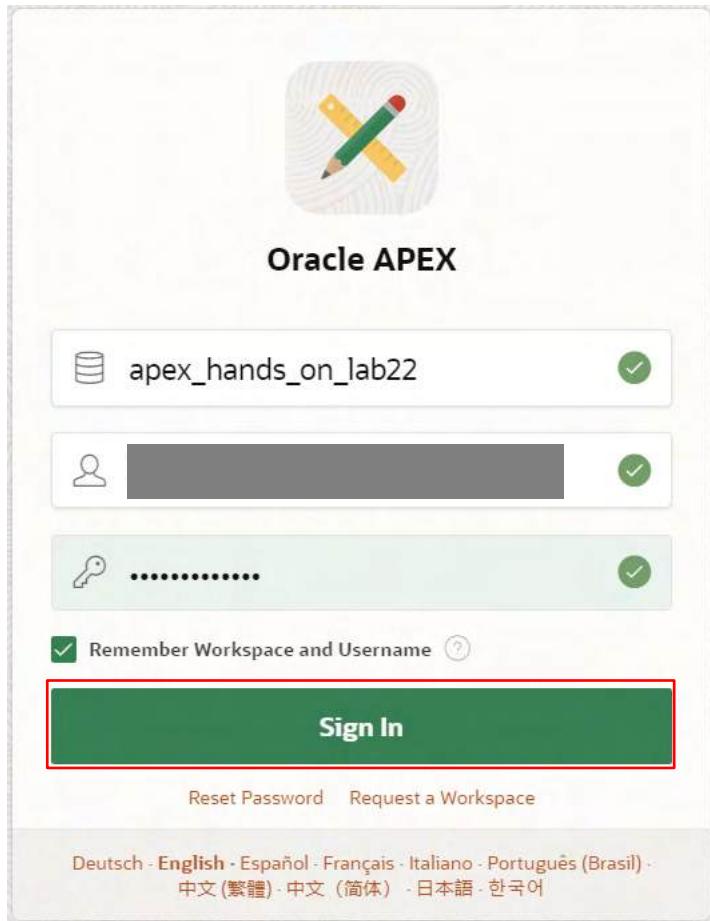
Tasks

Major Components of APEX

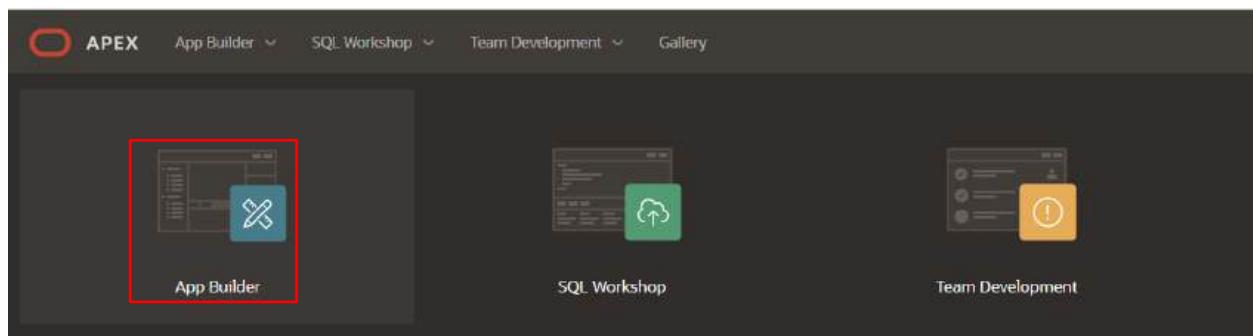
To log in to Oracle APEX, you need a Workspace Name, username, and the password created for that Workspace. In this hands-on lab, you log in to your Oracle APEX Workspace.

1. Log in to **Oracle APEX Workspace**. Perform the following steps:
 - a. Open your browser and enter the URL to sign in to the APEX development environment.

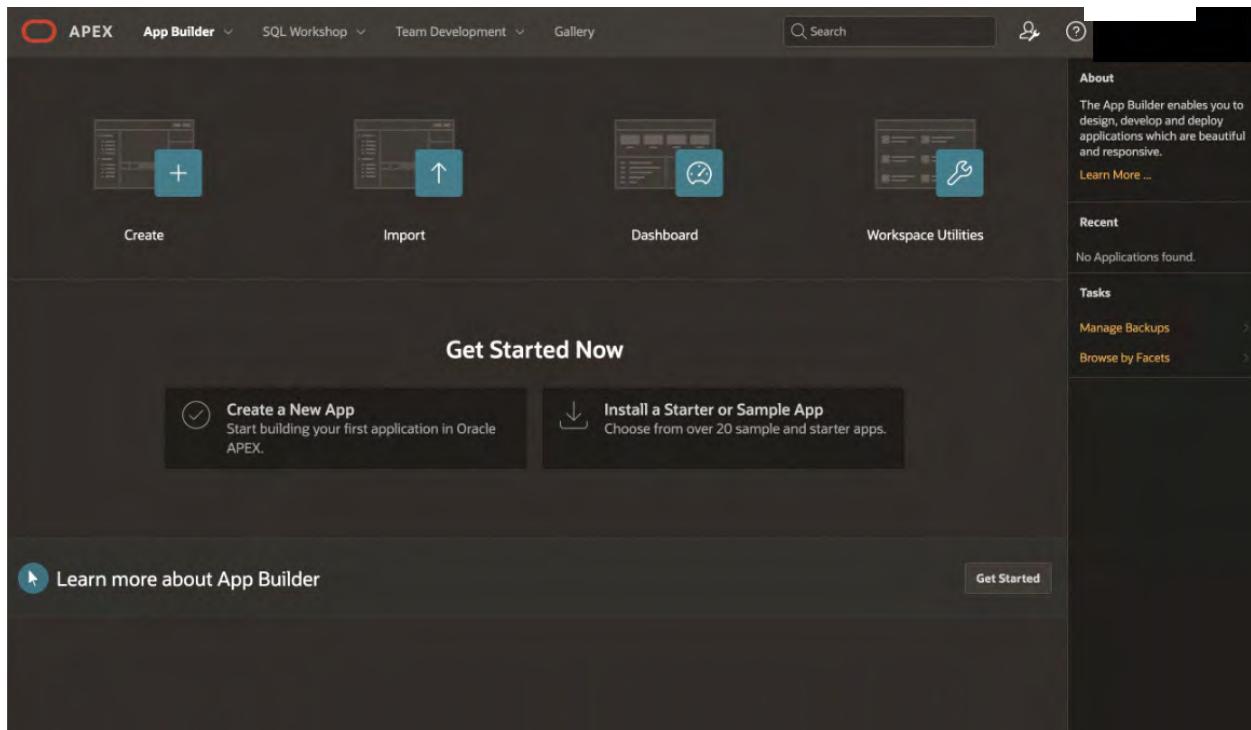
- b. The login page appears. Enter the workspace name, username, and password. Click **Sign In**.



2. The **Workspace** home page appears. Click **App Builder**.

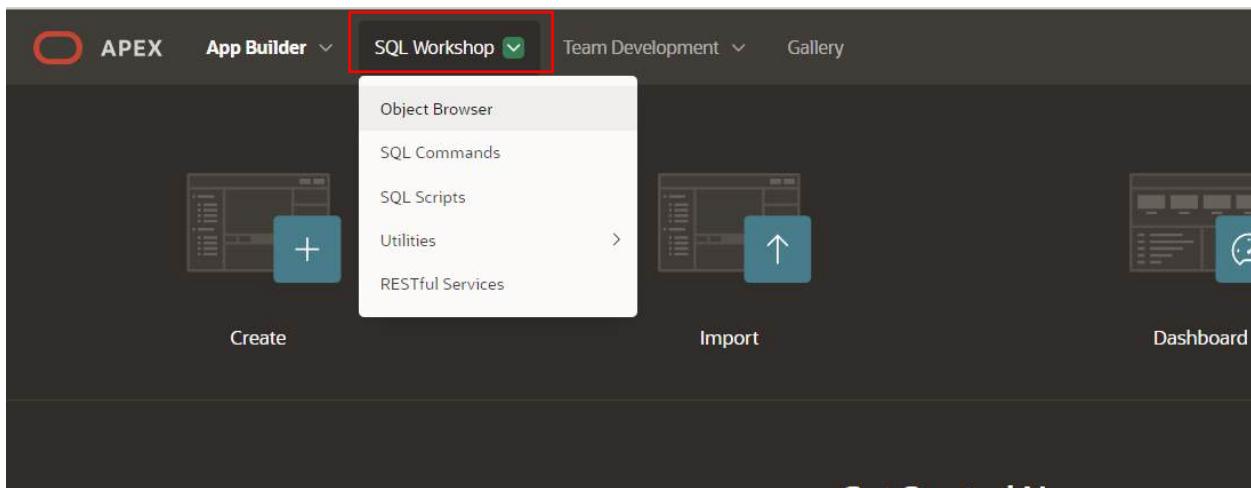


3. You do not have any applications listed right now. You can create a new app or install a **sample** or **starter app**.



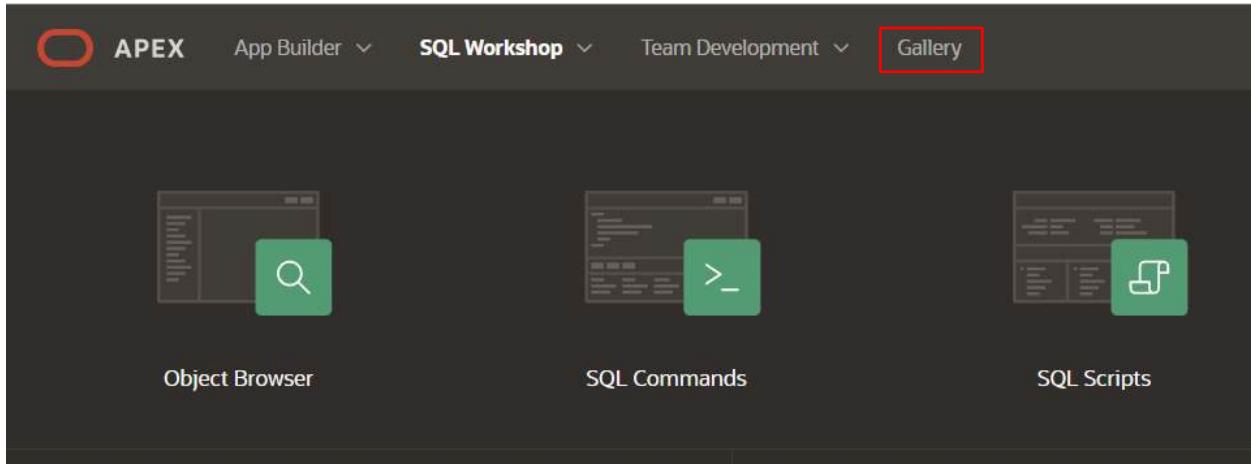
4. Click **SQL Workshop**.

The SQL Workshop home page appears. Review each of the **SQL Workshop** components.





5. Click **Gallery**.



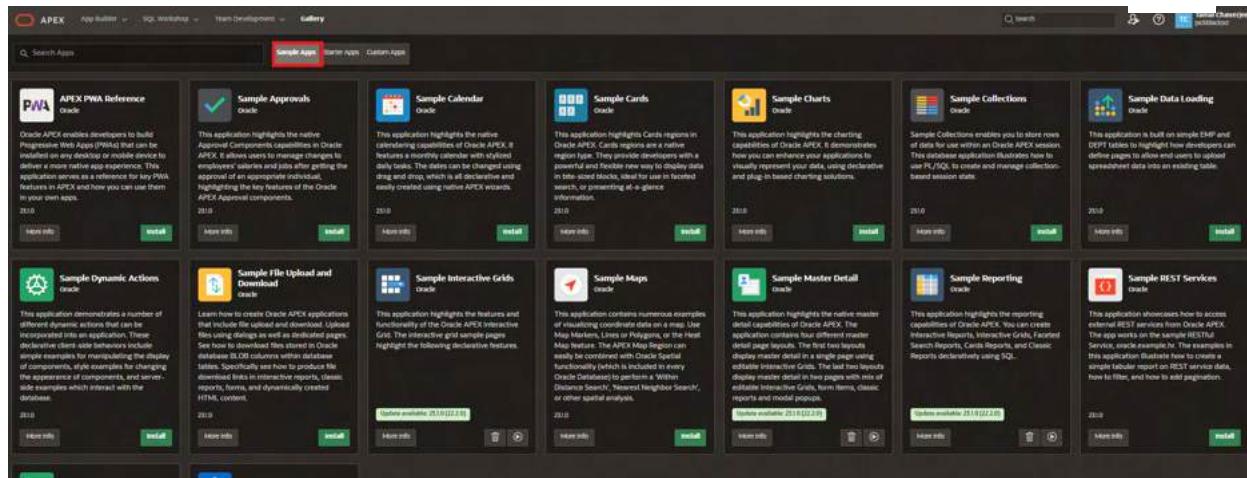
6. The Gallery page displays a collection of starter applications, sample applications and custom applications that you can install in your workspace.

| Sample Application | Description | Version | More Info | Install |
|---------------------------------|---|---------|---------------------------|-------------------------|
| APEx PWA Reference | This application highlights the native Progressive Web App (PWA) that can be installed on any desktop or mobile device to deliver a more native app experience. This application also provides references for key PWA features in APEx and how you can use them in your own apps. | 20.1.0 | More info | Install |
| Sample Approvals | This application highlights the native Approval Components capabilities in Oracle APEx. It allows users to manage changes to employees' salaries and jobs after getting the approval of their manager. | 20.1.0 | More info | Install |
| Sample Calendar | This application highlights the native Scheduling capabilities of Oracle APEx. It features a monthly calendar with stylized daily tasks. The dates can be changed using drag and drop, which is a fast, responsive and easy way to display data in a grid. | 20.1.0 | More info | Install |
| Sample Cards | This application highlights Card regions in Oracle APEx. Card regions are a native region type. They provide developers with a powerful and flexible new way to display data in a grid. Cards are blocks, used for user input, search, or presenting a gathering of a generic information. | 20.1.0 | More info | Install |
| Sample Charts | This application highlights the charting capabilities of Oracle APEx. It demonstrates how you can enhance your data, using decorative and plug-in based charting solutions. | 20.1.0 | More info | Install |
| Sample Collections | Sample Collections enables you to store more data for use within an Oracle APEx session. This database application illustrates how to use PL/SQL to create and manage collection-based session state. | 20.1.0 | More info | Install |
| Sample Data Loading | This application is built on simple EMP and DEPT tables to highlight how developers can define pages to allow end users to upload spreadsheet data into an existing table. | 20.1.0 | More info | Install |
| Sample Dynamic Actions | This application demonstrates a number of different dynamic actions that can be incorporated into an application. These include client-side validation, mouseover examples for manipulating the display of components, style examples for changing the appearance of components, and server-side examples which interact with the database. | 20.1.0 | More info | Install |
| Sample File Upload and Download | Learn how to create Oracle APEx applications that include file upload and download. Upload files using drag as well as dedicated pages. Download files using standard file download database BLOB columns within database tables. Specifically see how to produce the download links in interactive reports, classic reports, forms, and dynamically created HTML content. | 20.1.0 | More info | Install |
| Sample Interactive Grids | This application highlights the features and functionality of the Oracle APEx Interactive Grid. The interactive grid sample pages highlight the following decorative features: | 20.1.0 | More info | Install |
| Sample Maps | This application contains numerous examples of visualizing coordinate data on a map. Use Map Markers, Lines or Polygons, or the Heat Map feature to display data on a map. Data can easily be combined with Oracle Spatial functionality (which is included in every Oracle Database) to perform a "Within Distance Search", "Nearest Neighbor Search", or other spatial analysis. | 20.1.0 | More info | Install |
| Sample Master Detail | This application highlights the native master detail capabilities of Oracle APEx. The application contains four different master detail examples. One example shows how to display master detail in a single page using editable interactive Grids. The last two layouts display master detail in two pages with mix of interactive reports, classic reports, forms, items, classic reports and modal popups. | 20.1.0 | More info | Install |
| Sample Reporting | This application highlights the reporting capabilities of Oracle APEx. You can create Interactive Reports, Interactive Grids, Faceted Search Reports, Card Reports, and Classic Reports and modal popups. | 20.1.0 | More info | Install |
| Sample REST Services | This application showcases how to access external REST services from Oracle APEx. The app works on the sample RESTful Sample REST Service. The application shows how in this application (basically how to create a single tabular report on REST service data, how to filter, and how to add pagination). | 20.1.0 | More info | Install |

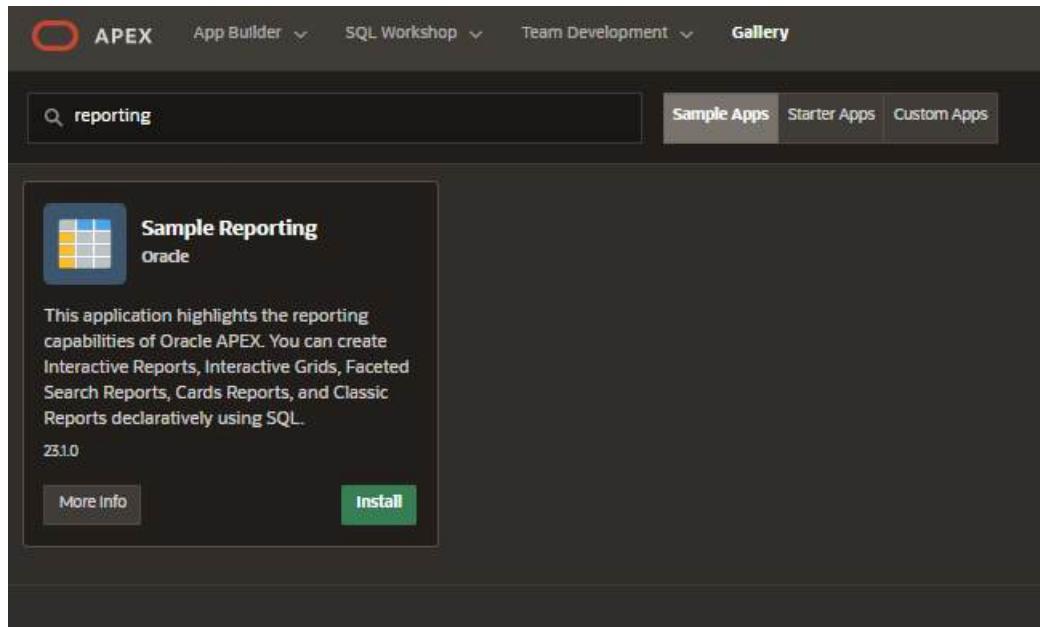
Install a Sample Application

Now, we will learn how to install and run a sample application.

- Once you navigate to Gallery, click **Sample Apps**.



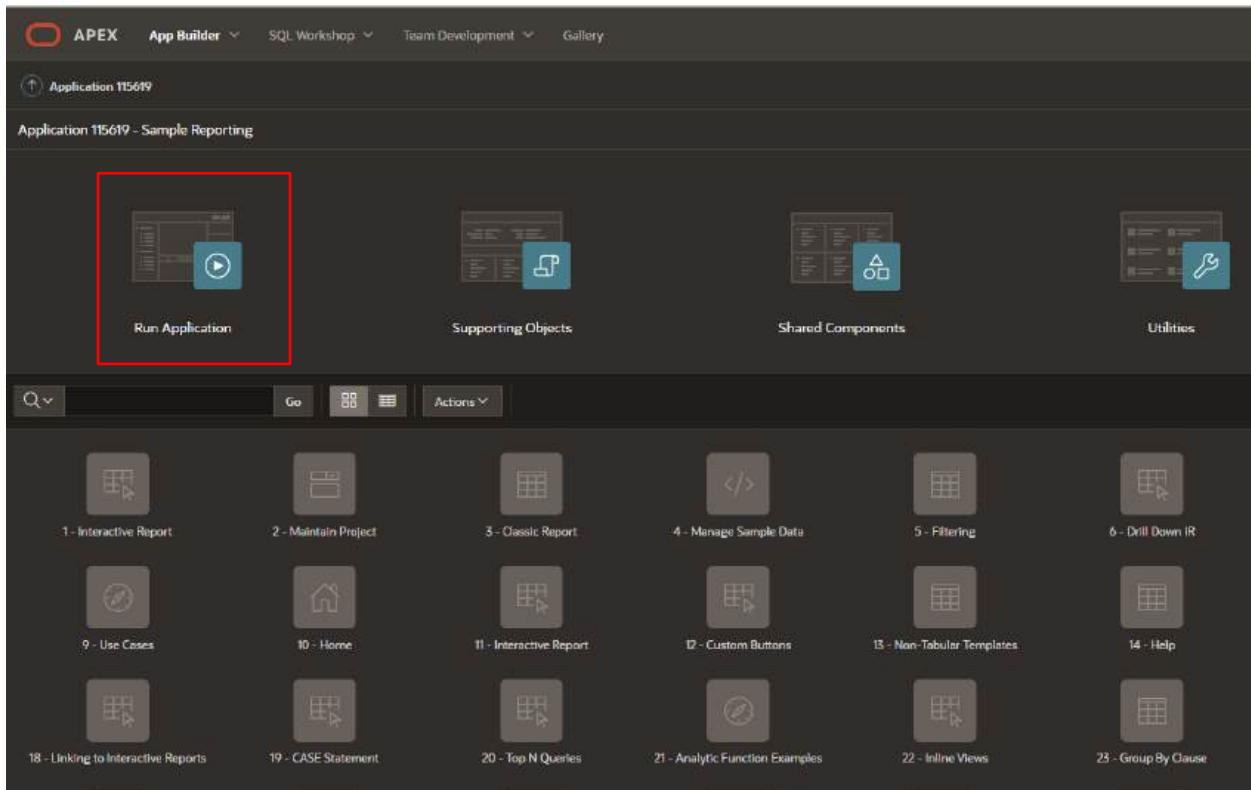
- Now that you are in **Sample Apps**, search for apps by typing "Reporting" into the "Search Apps" search box and navigate to **Sample Reporting** and click the **Install** button.



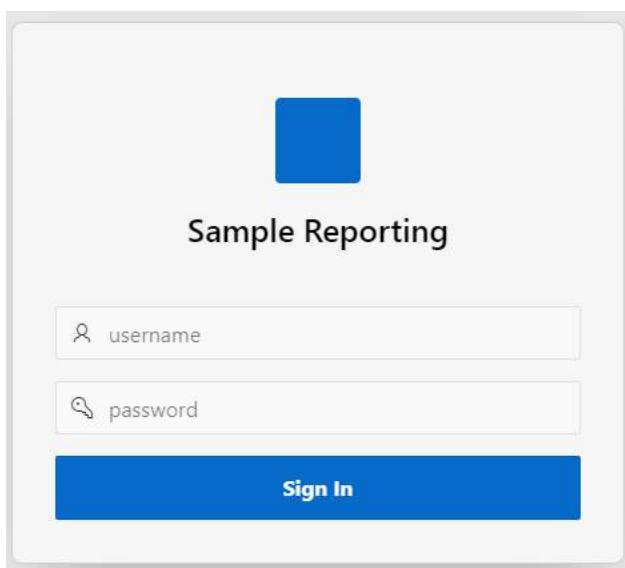
3. On the Install Application page, click the **Install Application** button.



4. You can now see that the application is installed. Run the application by selecting **Run Application**.



5. Log in to the **Sample Reporting** application as an end user.
6. Enter your **Username** and **Password** (Same as your Workspace credentials) and click **Sign In**.



7. You now see the Sample Reporting application home page. In your Runtime environment under **Developer Toolbar**, click **Home** to return to the home page of the APEX development environment.

The screenshot shows the 'Sample Reporting' application home page. On the left, there is a sidebar with a dark background containing links: 'Sample Reports' (Interactive Report, Interactive Grid, Faceted Search, Cards, Classic Report), 'Use Cases' (SQL Examples, Analytic Functions), and 'Administration'. The main content area has a light blue header 'Sample Reporting' with the subtext 'Demonstration of reports and reporting techniques in Oracle APEX'. Below the header is a section titled 'About' with a brief description: 'This application highlights the reporting capabilities of Oracle APEX. Use this application to better understand the native and declarative reporting functionality of APEX and how to write simple and advanced SQL.' There are four rows of cards, each with a title and a brief description:

| Interactive Report | Interactive Grid | Faceted Search | Cards |
|--|---|---|--|
| Reports which enable customization by the end user | Reports which allow for in-line editing and other customization by the end user | Report with facets to filter the result | Report displayed in cards |
| Classic Report | Use Cases | SQL Examples | Analytic Functions |
| Tabular data which can be filtered by page item values | Demonstrations of advanced APEX report techniques | Demonstrations of advanced SQL techniques | Demonstrations of various analytic functions available in Oracle SQL |

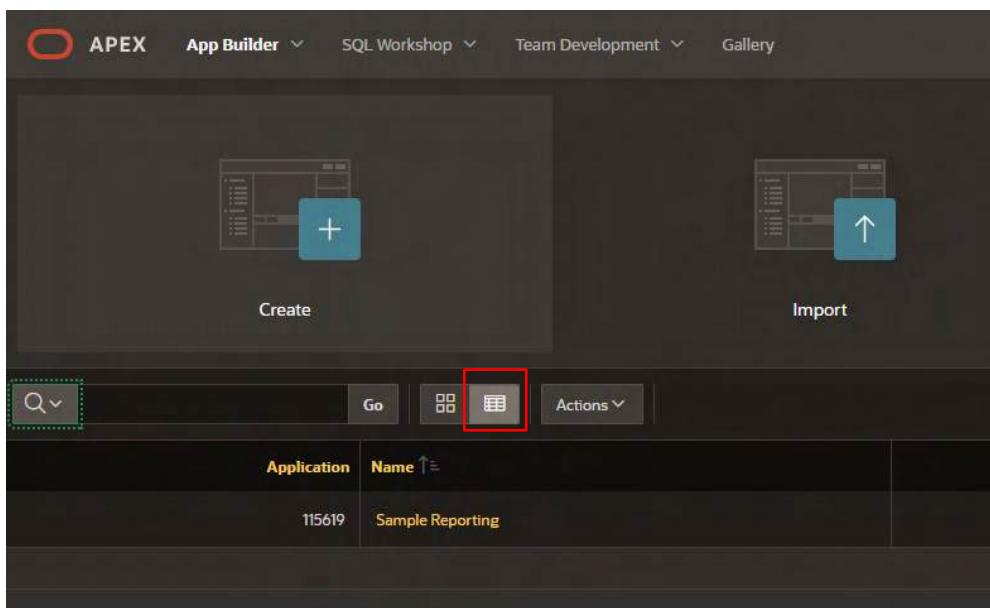
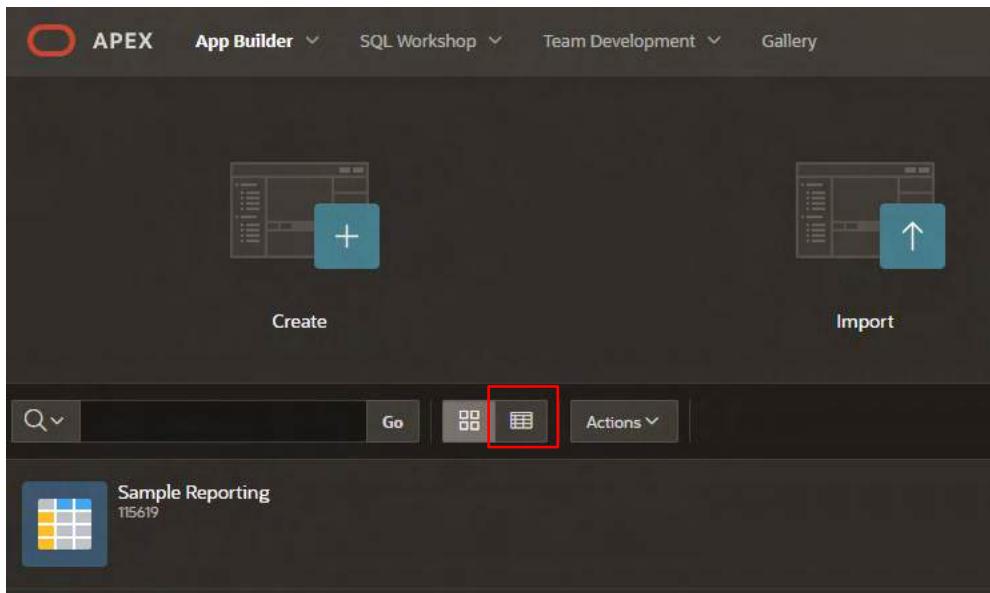
At the bottom, there are social media sharing icons for Twitter, LinkedIn, Facebook, and YouTube.

The screenshot shows the Oracle APEX developer toolbar. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, it says 'Application 115619'. The main content area displays four items: 'Run Application' (with a play icon), 'Supporting Objects' (with a database icon), 'Shared Components' (with a folder icon), and 'Utilities' (with a wrench icon). The 'App Builder' item is highlighted with a red box.

8. Click **App Builder**.

The screenshot shows the Oracle APEX developer toolbar again. The 'App Builder' icon is highlighted with a red box. Other items visible include 'SQL Workshop' and 'Gallery'.

9. Toggle between the **View Icons** and the **View Reports** buttons. You see that applications are displayed along with their icons and report format.



Summary

You now know how to navigate through the major components of Oracle APEX and install and run a packaged application. You may now **proceed to the next practice**.

Practice: Use SQL Workshop

Practice 1: Install a Sample Data Set

Overview

In this practice, you learn to install sample tables and views from Sample Datasets. This particular sample dataset is a collection of customers, stores, products, and orders used to manage the shopping cart.

In this practice, you will:

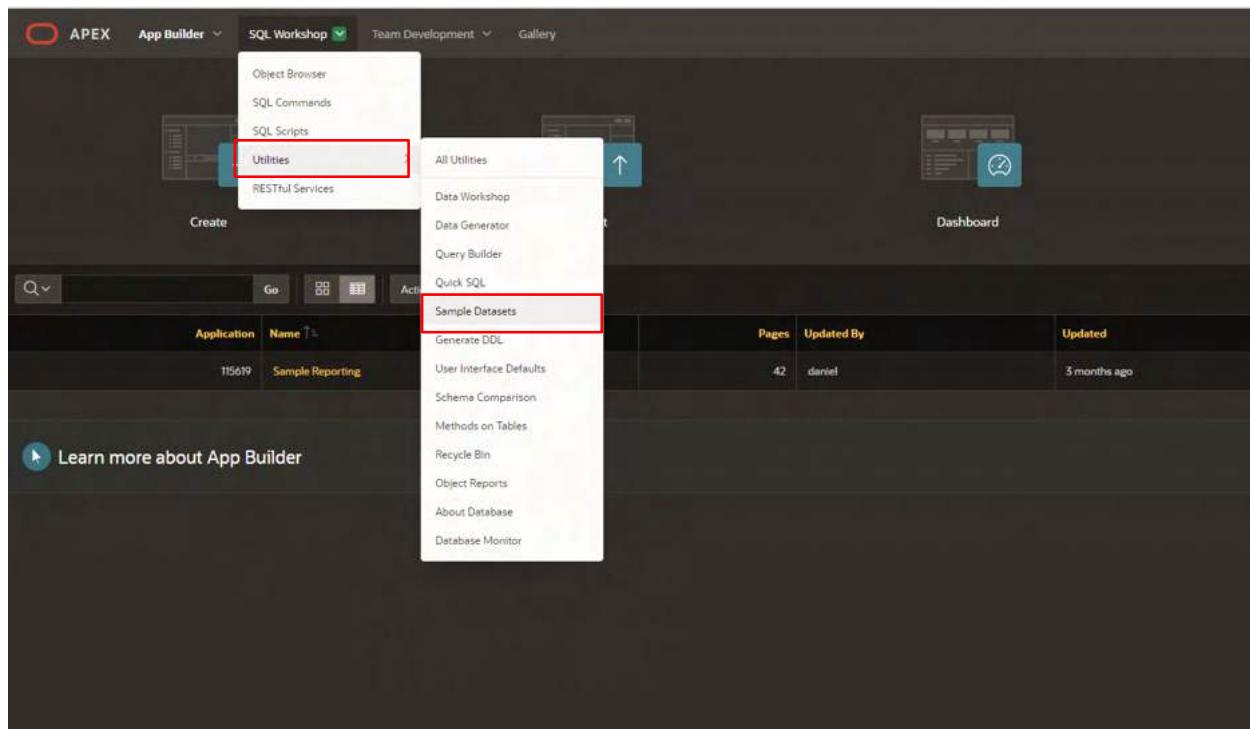
- Install a sample dataset into your Oracle APEX Workspace

Task

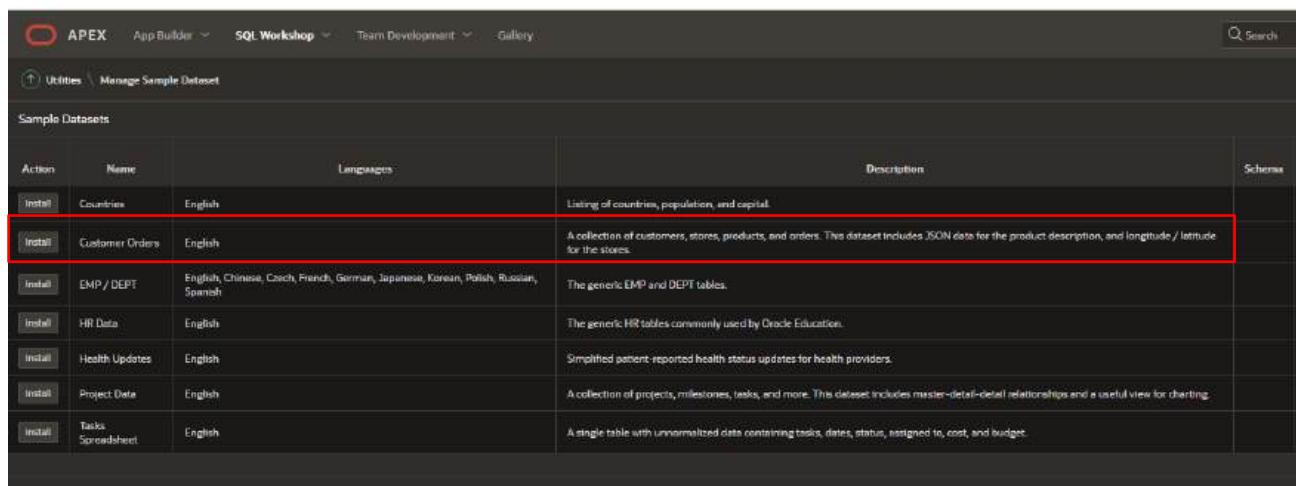
1. Log in to your workspace. If you are already logged in, click the **APEX** Logo in the upper left-hand corner of the page to return to the APEX workspace home page.

Create Customer Order Tables

2. From the APEX workspace home page, select the down arrow to the right of **SQL Workshop**, then select **Utilities** to display the options and choose **Sample Datasets**.



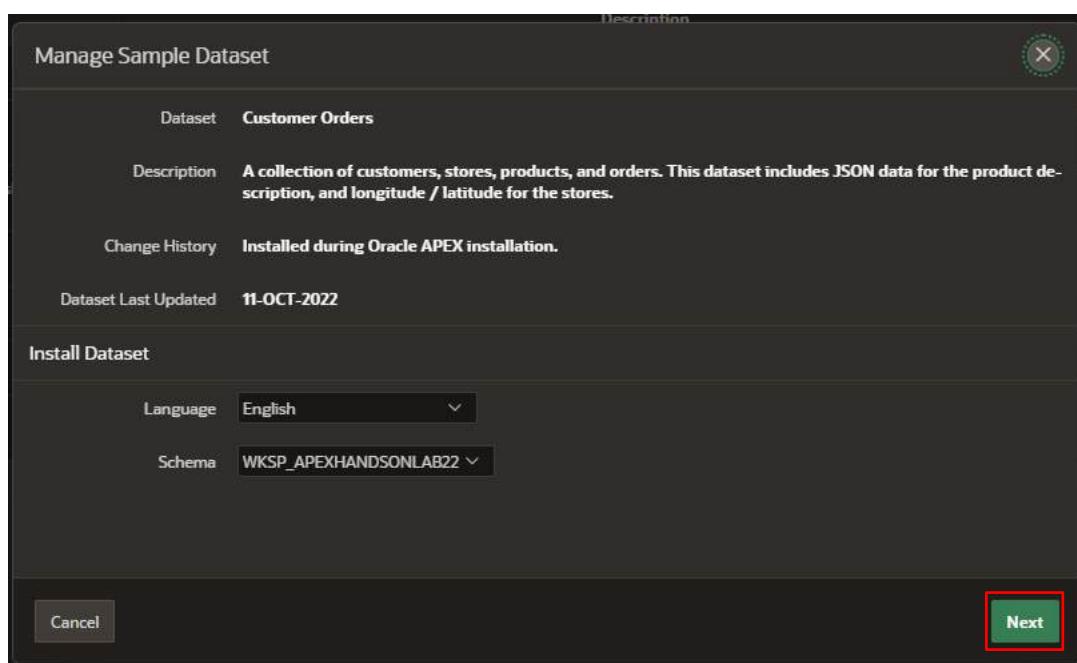
3. On the **Customer Orders** row, click **Install**.



| Action | Name | Languages | Description | Schema |
|---------|------------------------|---|--|--------|
| Install | Countries | English | Listing of countries, population, and capital. | |
| Install | Customer Orders | English | A collection of customers, stores, products, and orders. This dataset includes JSON data for the product description, and longitude / latitude for the stores. | |
| 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 | Health Updates | English | Simplified patient-reported health status updates for health providers. | |
| Install | Project Data | English | A collection of projects, milestones, tasks, and more. This dataset includes master-detail relationships and a useful view for charting. | |
| Install | Tasks Spreadsheet | English | A single table with unnormalized data containing tasks, dates, status, assigned to, cost, and budget. | |

4. Leave the settings to default and then click **Next**.

The schema name defaults to your current schema and hence will be different from the schema name shown below.

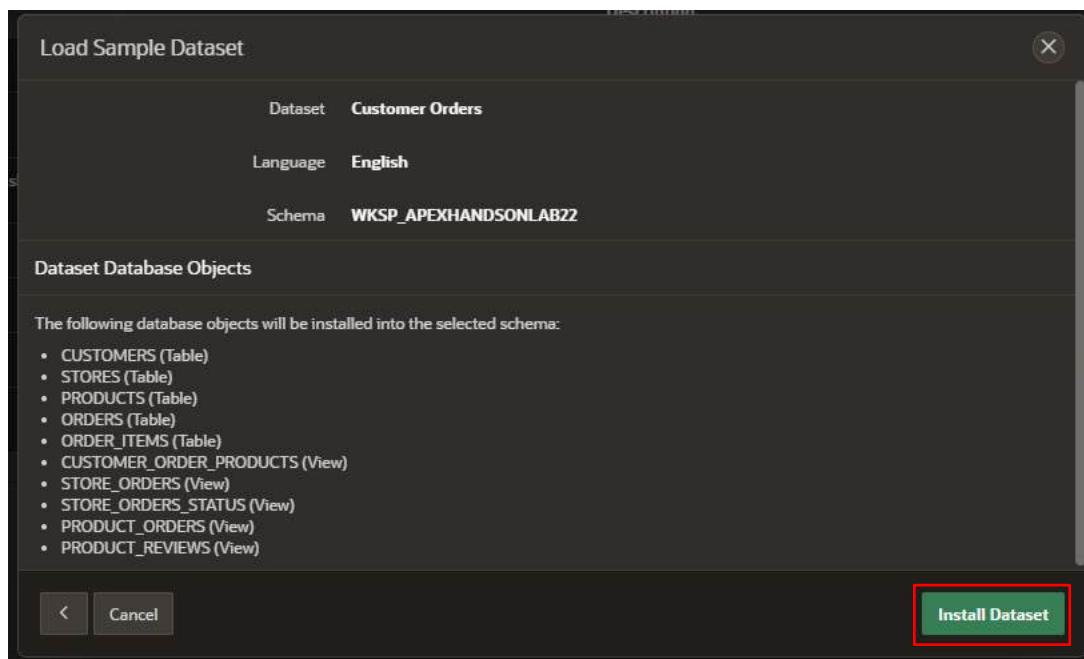


Manage Sample Dataset

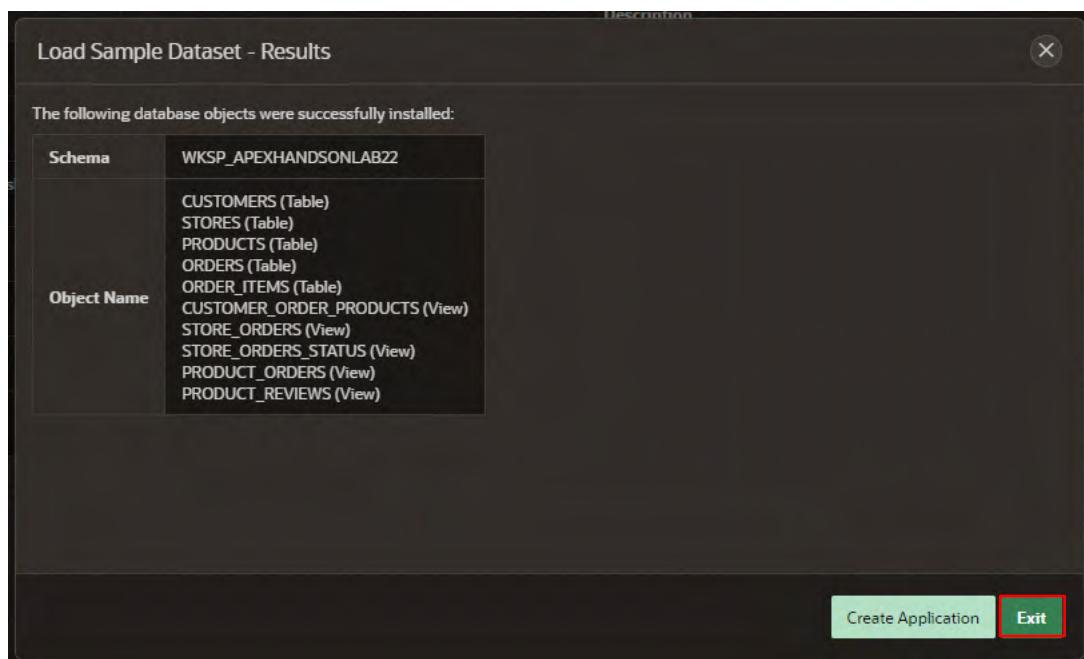
| | |
|----------------------|--|
| Dataset | Customer Orders |
| Description | A collection of customers, stores, products, and orders. This dataset includes JSON data for the product description, and longitude / latitude for the stores. |
| Change History | Installed during Oracle APEX installation. |
| Dataset Last Updated | 11-OCT-2022 |
| Install Dataset | |
| Language | English |
| Schema | WKSP_APEXHANDSONLAB22 |

Cancel **Next**

5. Click **Install Dataset**.



6. Click **Exit**.



You now know how to install sample tables and views from Sample Datasets. You may now **proceed to the next practice**.

Practice 2: Add Columns to the PRODUCTS Table

Overview

The **PRODUCTS** table includes some columns such as image, price, and details. But there are other characteristics that customers would appreciate knowing about a product, such as color, type of clothing, and department. In this practice, you will add these columns to the **PRODUCTS** table.

Since many products share the same colors, clothing type, and department, to avoid data redundancy it is a best practice to create three additional tables to store the unique values of color, type, and department data. Instead of creating these three tables for yourself, you'll use the **Create Lookup Table** feature.

In this practice, you learn how to add these three new columns to the **PRODUCTS** table and then create lookup tables for those new columns.

In this practice, you will:

- Add new columns to the existing **PRODUCTS** table
- Populate the new columns
- Create lookup tables

Tasks

Add Column Title to the **PRODUCTS** Table

1. From the APEX Main Menu, click **SQL Workshop**.

2. Click Object Browser.

The screenshot shows the Oracle APEX interface with the 'SQL Workshop' tab selected. A context menu is open over a table row in the 'Sample Datasets' section. The menu items are: Utilities, Manage Sample Dataset, Object Browser (which is highlighted with a red box), SQL Commands, SQL Scripts, Utilities, and RESTful Services.

| Action | Name | Description |
|---------|-------------------|--|
| Install | Countries | Listing of countries, population, and capital. |
| Update | Customer Orders | A collection of customers, stores, products, and orders. This longitude / latitude for the stores. |
| Install | EMP / DEPT | The generic EMP and DEPT tables. |
| Install | HR Data | The generic HR tables commonly used by Oracle Education. |
| Install | Health Updates | Simplified patient-reported health status updates for health. |
| Install | Project Data | A collection of projects, milestones, tasks, and more. This da for charting. |
| Install | Tasks Spreadsheet | A single table with unnormalized data containing tasks, date |

3. Navigate to PRODUCTS. Click the Add Column button.

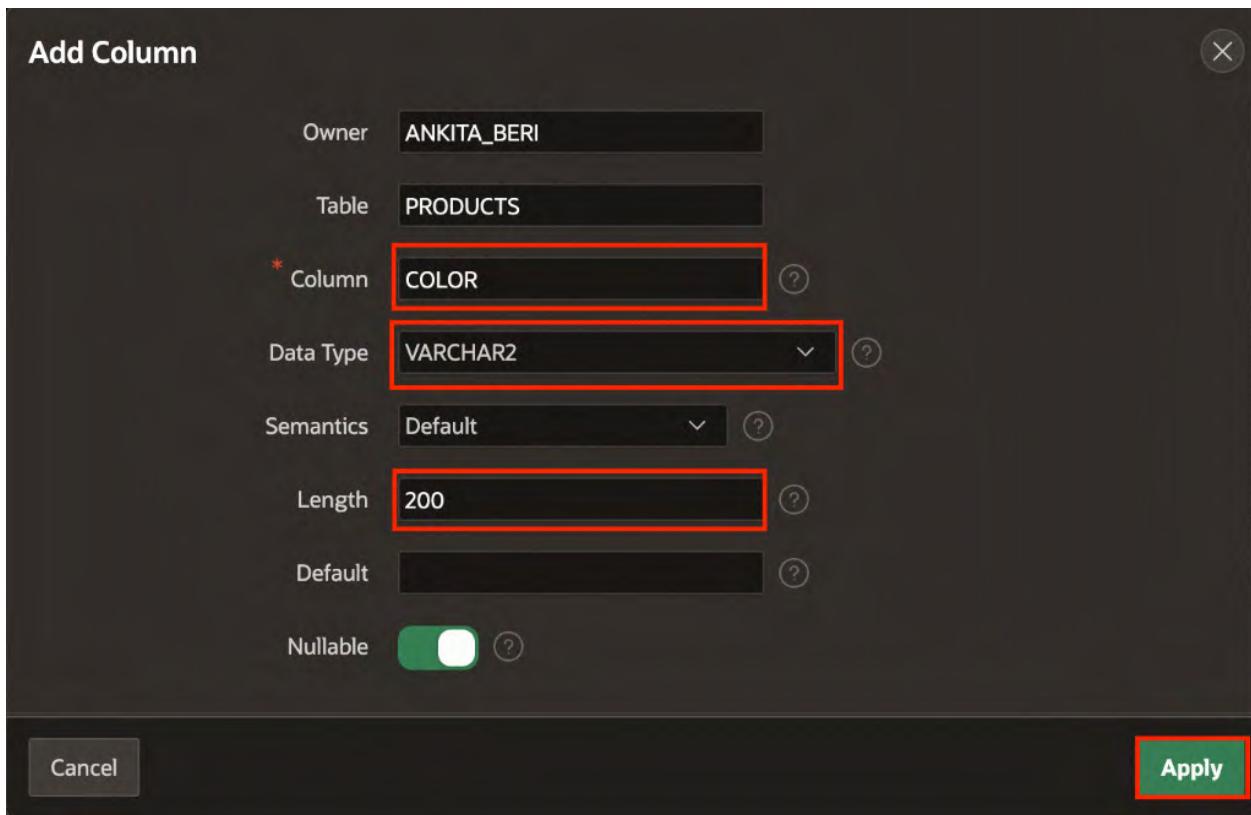
The screenshot shows the Oracle APEX Object Browser. The left sidebar shows various database objects like Tables, Views, Indexes, Sequences, Types, and Packages. The 'Tables' section is expanded, and the 'PRODUCTS' table is selected (highlighted with a green box). The main panel displays the columns of the 'PRODUCTS' table. The 'Add Column' button at the top of the table list is highlighted with a red box.

| Column Name | Data Type | Nullable | Default | Primary Key | Comment.. | Identity |
|--------------------|--------------------|----------|-----------|-------------|------------|-----------|
| PRODUCT_ID | NUMBER | N | 'ANKIT... | 1 | Auto-i... | BY DEF... |
| PRODUCT_NAME | VARCHAR2(255 BYTE) | N | | | What a... | |
| UNIT_PRICE | NUMBER(10,2) | Y | | | The m... | |
| PRODUCT_DETAILS | BLOB | Y | | | Furthe... | |
| PRODUCT_IMAGE | BLOB | Y | | | A pictu... | |
| IMAGE_MIME_TYPE | VARCHAR2(512 BYTE) | Y | | | The mi... | |
| IMAGE_FILENAME | VARCHAR2(512 BYTE) | Y | | | The na... | |
| IMAGE_CHARSET | VARCHAR2(512 BYTE) | Y | | | The ch... | |
| IMAGE_LAST_UPDATED | DATE | Y | | | The da... | |

4. For the Color column, enter the following:

- Add Column - enter **COLOR**
- Type - select **VARCHAR2**
- Length - enter **200**

Click **Apply**.



5. Click the **Add Column** button.
6. For the Department column, enter the following:
 - Add Column - enter **DEPARTMENT**
 - Type - select **VARCHAR2**
 - Length - enter **200**.

Click **Apply**.

Add Column

| | |
|--|-------------------------------------|
| Owner | ANKITA_BERI |
| Table | PRODUCTS |
| * Column | DEPARTMENT |
| Data Type | VARCHAR2 |
| Semantics | Default |
| Length | 200 |
| Default | |
| Nullable | <input checked="" type="checkbox"/> |
| <input type="button" value="Cancel"/> <input type="button" value="Apply"/> | |

7. Click the **Add Column** button.

APEX App Builder SQL Workshop Team Development Gallery Search

Object Browser

Tables CUSTOMERS EBA_DEMO_IR_DEPT EBA_DEMO_IR_EMP EBA_DEMO_IR_PROJECTS ORDERS ORDER_ITEMS PRODUCTS STORES

PRODUCTS

| PRODUCTS | | | |
|---|---------------|----------|---|
| Table | Data | Indexes | Model |
| Add Column Modify Column Rename Column Drop Column Rename Copy Drop Truncate Create Lookup Table Create App | | | |
| Column Name | Data Type | Nullable | Default |
| PRODUCT_ID | NUMBER | No | "WKSP_APEXHANDSONLAB22"."SEQ\$\$.NEXTVAL" |
| PRODUCT_NAME | VARCHAR2(255) | No | - |
| UNIT_PRICE | NUMBER(10,2) | Yes | - |
| PRODUCT_DETAILS | BLOB | Yes | - |
| PRODUCT_IMAGE | BLOB | Yes | - |
| IMAGE_MIME_TYPE | VARCHAR2(512) | Yes | - |
| IMAGE_FILENAME | VARCHAR2(512) | Yes | - |
| IMAGE_CHARSET | VARCHAR2(512) | Yes | - |
| IMAGE_LAST_UPDATED | DATE | Yes | - |
| COLOR | VARCHAR2(200) | Yes | - |
| DEPARTMENT | VARCHAR2(200) | Yes | - |

Download | Print

8. For the Clothing column, enter the following:

- Add Column - enter **CLOTHING**
- Type - select **VARCHAR2**
- Length - enter **200**.

Click **Next**.

Add Column

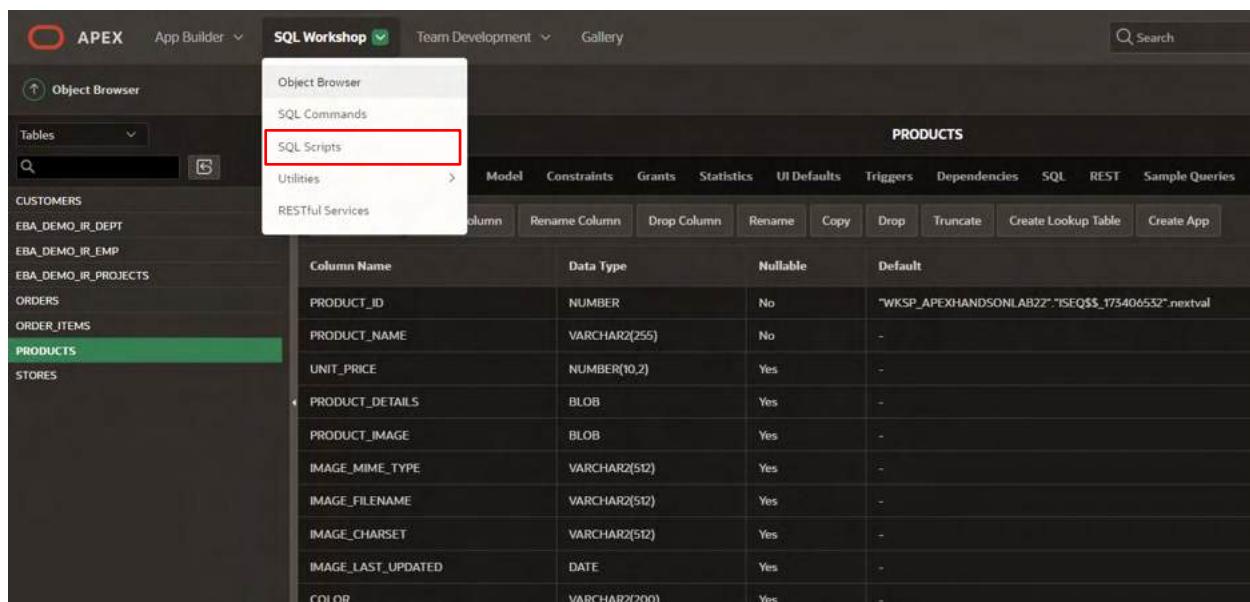
| | |
|-----------|-------------------------------------|
| Owner | ANKITA_BERI |
| Table | PRODUCTS |
| * Column | CLOTHING |
| Data Type | VARCHAR2 |
| Semantics | Default |
| Length | 200 |
| Default | |
| Nullable | <input checked="" type="checkbox"/> |

Cancel **Apply**

Populate New Columns

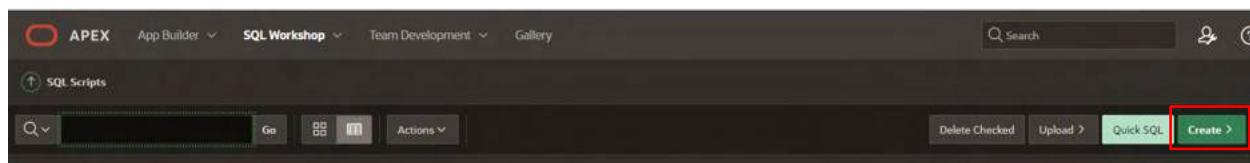
1. From the APEX main menu, click **SQL Workshop**.

2. Click **SQL Scripts**.



The screenshot shows the Oracle APEX interface with the SQL Workshop module active. In the left sidebar, under the Object Browser, the 'Tables' section is open, showing various database tables like CUSTOMERS, ORDERS, and PRODUCTS. The 'PRODUCTS' table is currently selected, indicated by a green highlight. In the top navigation bar, the 'SQL Workshop' tab is selected. A dropdown menu is open over the 'SQL Scripts' option, which is also highlighted with a red box. The main workspace displays the structure of the 'PRODUCTS' table, including columns such as PRODUCT_ID, PRODUCT_NAME, UNIT_PRICE, and COLOR, along with their data types and constraints.

3. Click **Create**.



The screenshot shows the Oracle APEX interface with the SQL Scripts module active. The top navigation bar includes the 'SQL Workshop' tab. The main area shows a toolbar with various icons and a search bar. On the right side, there are buttons for 'Delete Checked', 'Upload >', 'Quick SQL', and a prominent green 'Create >' button, which is highlighted with a red box. This screen allows users to manage and create SQL scripts.

4. For Script Name, enter **Populating new columns**.

5. Copy the following script and paste it into the editor.

```
UPDATE
(
    SELECT p.product_id,
           p.product_name,
           p.clothing,
           p.color,
           p.department,
           p.product_details
      FROM products p ) p
SET    p.clothing = Substr(product_name, Instr(product_name, ',',1,1)+1, Instr(product_name, ',',1, 2)+1 - Instr(product_name, ',',1,1)- 2),
       p.color =
(
    SELECT c.color
```

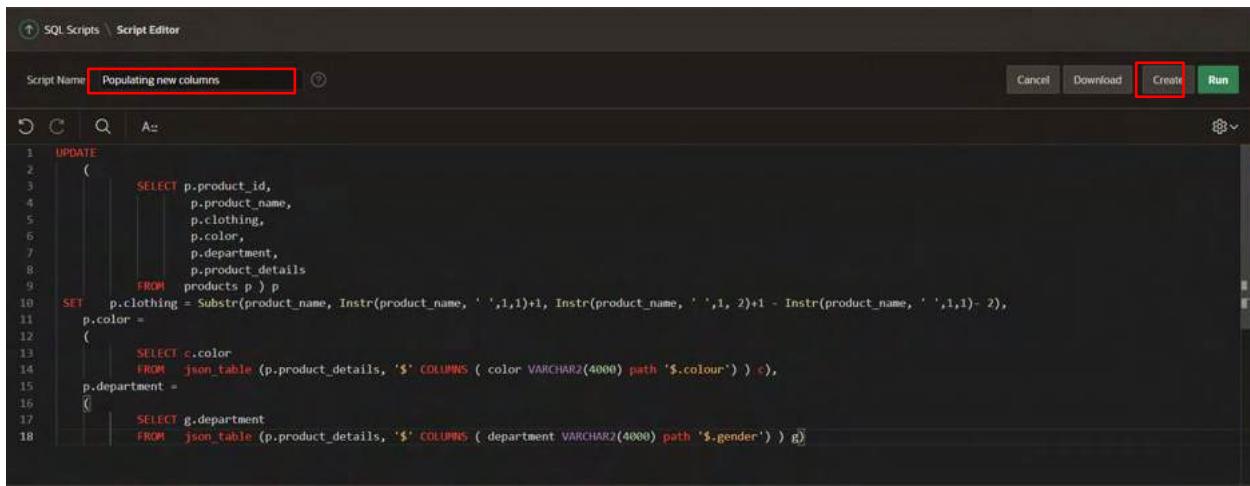
```

        FROM json_table (p.product_details, '$' COLUMNS (
color VARCHAR2(4000) path '$.colour') ) c,
p.department =
(
    SELECT g.department
    FROM json_table (p.product_details, '$' COLUMNS (
department VARCHAR2(4000) path '$.gender') ) g)

```

This script inserts the unique product type values (e.g. Shirt, Jacket, Skirt, etc.) into the CLOTHING column in the **PRODUCTS** table. Similarly, it inserts the unique department names (e.g. Boys', Girls', Men's, Women's) and color names into the DEPARTMENT and COLOR columns, respectively, based on information found in the JSON product details column in the **PRODUCTS** table.

6. Click Run.



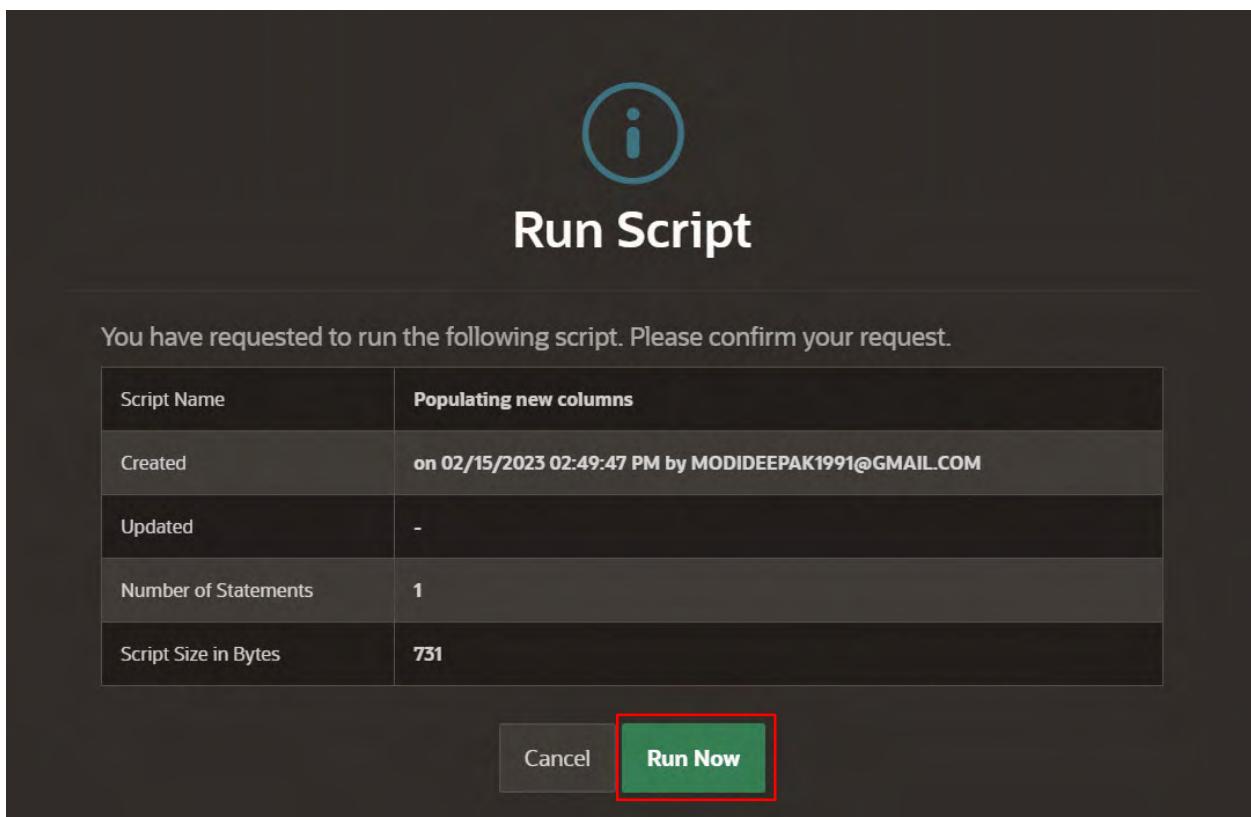
The screenshot shows the Oracle SQL Developer interface. The title bar says "SQL Scripts - Script Editor". The script name "Populating new columns" is highlighted with a red box. The toolbar has buttons for "Cancel", "Download", "Create", and "Run", with "Run" also highlighted with a red box. The main area contains the following PL/SQL code:

```

1 UPDATE
2   (
3     SELECT p.product_id,
4            p.product_name,
5            p.clothing,
6            p.color,
7            p.department,
8            p.product_details
9       FROM products p ) p
10  SET    p.clothing = Substr(product_name, Instr(product_name, ',' ,1,1)+1, Instr(product_name, ',' ,1, 2)+1 - Instr(product_name, ',' ,1,1)- 2),
11      p.color =
12      (
13        SELECT c.color
14        FROM json_table (p.product_details, '$' COLUMNS ( color VARCHAR2(4000) path '$.colour') ) c),
15      p.department =
16      (
17        SELECT g.department
18        FROM json_table (p.product_details, '$' COLUMNS ( department VARCHAR2(4000) path '$.gender') ) g)

```

7. Click **Run Now**.



8. The Script Results page is displayed listing the statements processed, successful, and with errors.

The screenshot shows the "Script Results" page from the SQL Workshop. The title bar indicates the script name is "Populating new columns" and the status is "Complete". The results table has columns: Number, Elapsed, Statement, Feedback, and Rows. One row is shown: "1", "0.08", "UPDATE (SELECT p.product_id, ...)", "46 row(s) updated.", "46". Below the table, summary statistics are provided: "Statements Processed: 1", "Successful: 1", and "With Errors: 0".

| Number | Elapsed | Statement | Feedback | Rows |
|--------|---------|------------------------------------|--------------------|------|
| 1 | 0.08 | UPDATE (SELECT p.product_id, ...) | 46 row(s) updated. | 46 |

Download row(s) 1 - 1 of 1

1 Statements Processed 1 Successful 0 With Errors

9. To check the values in the **PRODUCTS** table, click **SQL Workshop** and click **SQL Commands**.

The screenshot shows the Oracle APEX interface with the 'SQL Workshop' tab selected. In the center, there's a table with one row showing an UPDATE statement. On the left, there's a sidebar with a 'Script' section titled 'Populating new columns'. A context menu is open over the 'Results' link, listing 'Object Browser', 'SQL Commands' (which is highlighted with a red box), 'SQL Scripts', 'Utilities', and 'RESTful Services'.

10. Copy the following SQL query.

```
SELECT p.product_name,
       p.unit_price,
       p.color,
       p.department,
       p.clothing
  FROM products p;
```

11. Click **Run**.

The screenshot shows the 'SQL Commands' page in Oracle APEX. A SQL command is entered in the editor area:

```
1 SELECT p.product_name,
2        p.unit_price,
3        p.color,
4        p.department,
5        p.clothing
6   FROM products p;
```

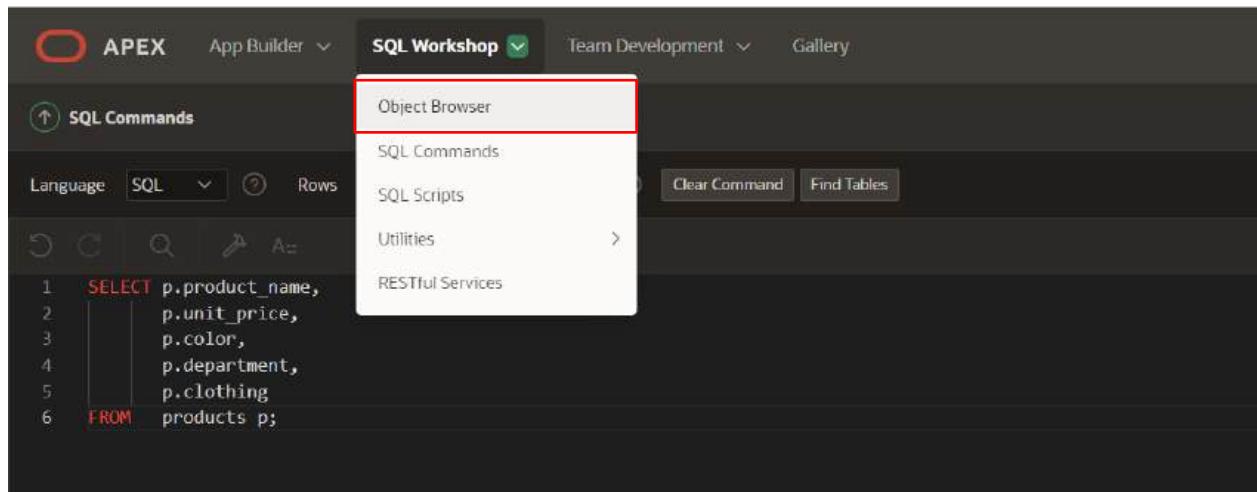
The 'Run' button at the top right of the editor is highlighted with a green box.

Create Lookup Tables

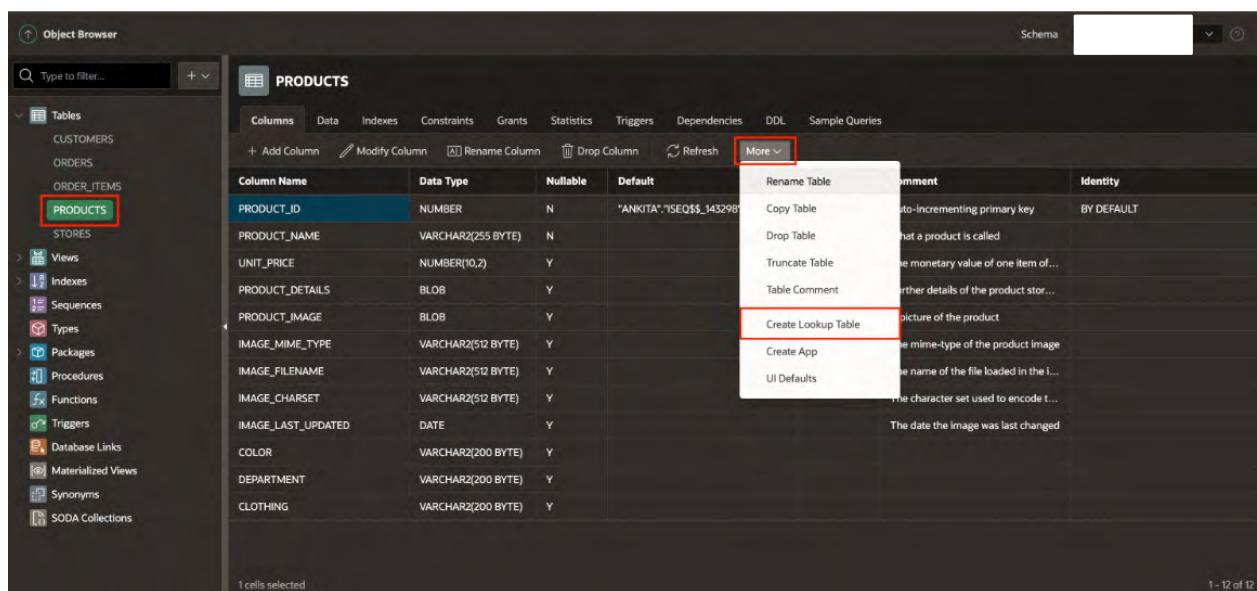
Since multiple products may have the same values for Color, Department, and Clothing, to avoid repetition and make updates easy, you can create a lookup table for each. A lookup table stores the value of the available colors, departments, or clothing in a single place, and then each product can reference the value from the lookup table.

In this practice, you create lookup tables based on the new three columns. After you create a lookup table, this allows the PRODUCTS table to reference the new color, clothing type, and department lookup value by its new numeric primary key value in the lookup table.

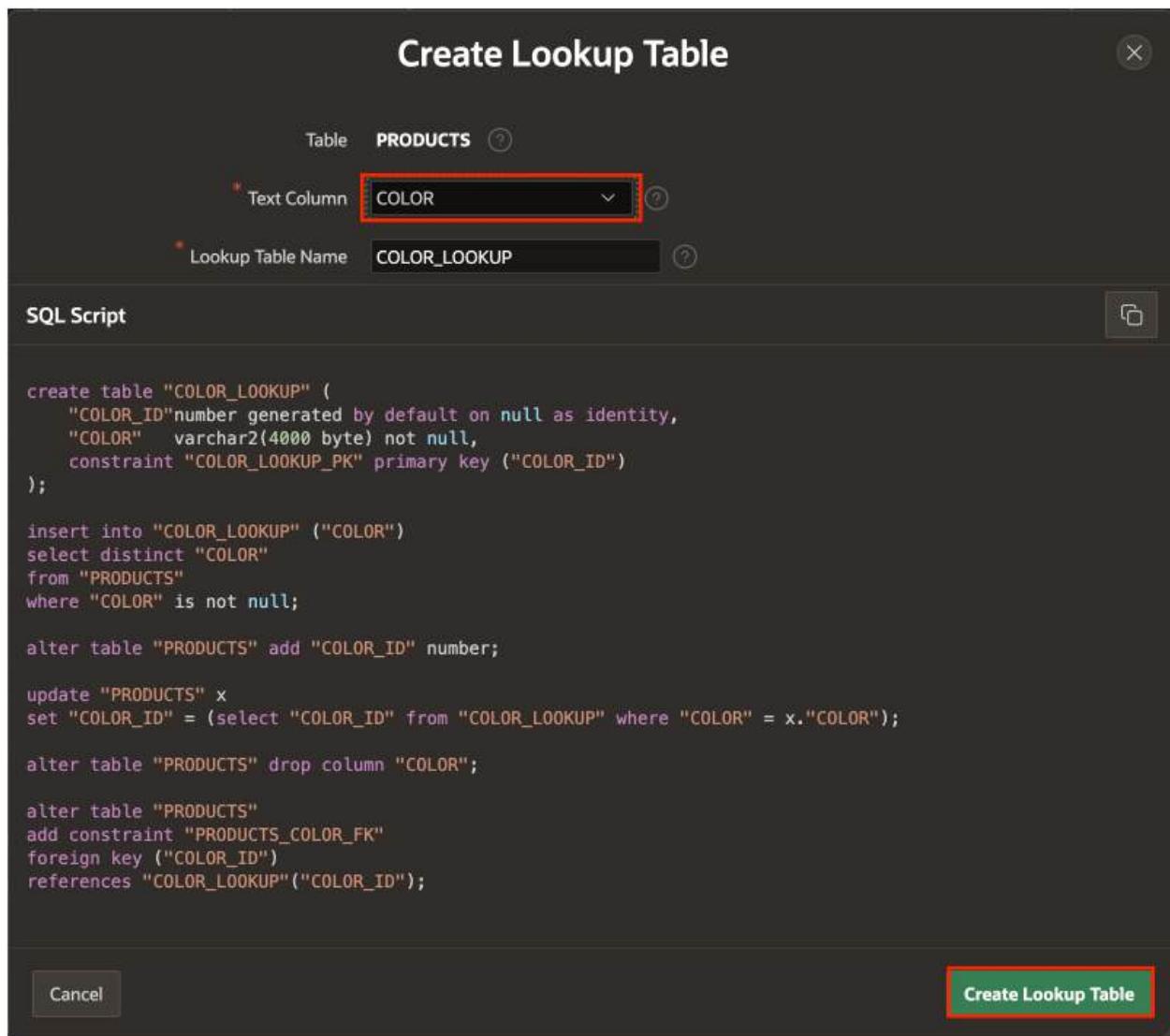
1. From the APEX main menu, click **SQL Workshop**.
2. Click **Object Browser**.



3. Navigate to the PRODUCTS table.
4. Click the **Create Lookup Table** button.



5. For **Text Column**, select **COLOR** and Click **Create Lookup Table**.

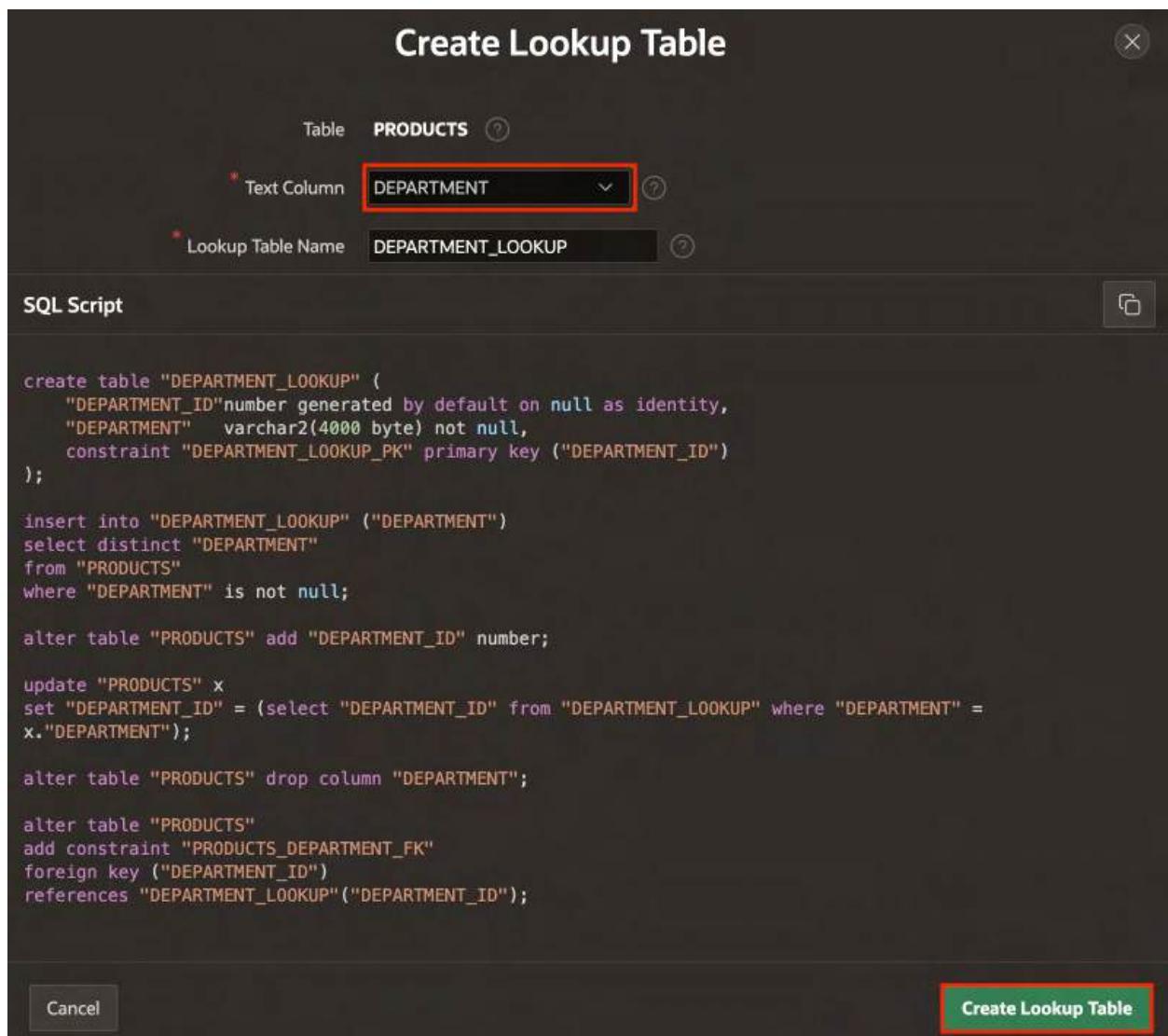


Note: Click the Create Lookup Table button only once. Then you will find the new table listed in the Object Browser.

6. To create **Department** lookup table, navigate back to the **Products** table and Click **More** and Select **Create Lookup Table** button.

The screenshot shows the Oracle SQL Workshop Object Browser. On the left, there's a tree view of database objects under the schema ANKITA. The 'Tables' node is expanded, and the 'PRODUCTS' table is selected, highlighted with a red box. In the main pane, the 'PRODUCTS' table is displayed with columns: PRODUCT_ID, PRODUCT_NAME, UNIT_PRICE, PRODUCT_DETAILS, PRODUCT_IMAGE, IMAGE_MIME_TYPE, IMAGE_FILENAME, IMAGE_CHARSET, IMAGE_LAST_UPDATED, COLOR, DEPARTMENT, and CLOTHING. The 'More' button in the toolbar is also highlighted with a red box. A context menu is open at the bottom right of the table area, listing options like Rename Table, Copy Table, Drop Table, Truncate Table, Table Comment, Create App, and UI Defaults. The 'Create Lookup Table' option is specifically highlighted with a red box. At the bottom of the table pane, it says '1 cells selected' and '1 - 12 of 12'.

7. For Text Column, select **DEPARTMENT** and Click **Create Lookup Table**.



*Note: Click the **Create Lookup Table** button only once. Then you will find the new table listed in the Object Browser.*

8. To create **Clothing** lookup table, navigate back to the **Products** table and Click **More** and Select **Create Lookup Table** button.

The screenshot shows the Oracle SQL Workshop interface. On the left is a sidebar with various database objects like Tables, Views, Indexes, etc. The main area shows the 'PRODUCTS' table with columns: PRODUCT_ID, PRODUCT_NAME, UNIT_PRICE, PRODUCT_DETAILS, PRODUCT_IMAGE, IMAGE_MIME_TYPE, IMAGE_FILENAME, IMAGE_CHARSET, IMAGE_LAST_UPDATED, CLOTHING, COLOR_ID, and DEPARTMENT_ID. The 'CLOTHING' column is currently selected. A context menu is open at the top right, with the 'More' option expanded. The 'Create Lookup Table' option is highlighted with a red box.

- For Text Column, select **CLOTHING** and Click **Create Lookup Table**.

The screenshot shows the 'Create Lookup Table' dialog. At the top, it says 'Create Lookup Table' and has a 'Table' dropdown set to 'PRODUCTS'. Below that, there are two fields: 'Text Column' with a dropdown containing 'CLOTHING' (highlighted with a red box) and 'Lookup Table Name' with the value 'CLOTHING_LOOKUP'. Underneath, the 'SQL Script' tab is active, showing the generated SQL code:

```

create table "CLOTHING_LOOKUP" (
    "CLOTHING_ID" number generated by default on null as identity,
    "CLOTHING" varchar2(4000 byte) not null,
    constraint "CLOTHING_LOOKUP_PK" primary key ("CLOTHING_ID")
);

insert into "CLOTHING_LOOKUP" ("CLOTHING")
select distinct "CLOTHING"
from "PRODUCTS"
where "CLOTHING" is not null;

alter table "PRODUCTS" add "CLOTHING_ID" number;

update "PRODUCTS" x
set "CLOTHING_ID" = (select "CLOTHING_ID" from "CLOTHING_LOOKUP" where "CLOTHING" = x."CLOTHING");

alter table "PRODUCTS" drop column "CLOTHING";

alter table "PRODUCTS"
add constraint "PRODUCTS_CLOTHING_FK"
foreign key ("CLOTHING_ID")
references "CLOTHING_LOOKUP"("CLOTHING_ID");

```

At the bottom, there are 'Cancel' and 'Create Lookup Table' buttons, with 'Create Lookup Table' highlighted with a red box.

Note: Click the **Create Lookup Table** button only once. Then you will find the new table listed in the Object Browser.

10. The columns COLOR, DEPARTMENT, and CLOTHING in the **Products** table are renamed to COLOR_ID, DEPARTMENT_ID, and CLOTHING_ID respectively, and their data type changed to NUMBER. Also, there are new tables containing the values of the products:
- COLOR_LOOKUP
 - DEPARTMENT_LOOKUP
 - CLOTHING_LOOKUP

The numeric value of the COLOR_ID column will now store a reference to the system assigned unique ID of a particular color, whose name is stored in the new COLOR_LOOKUP table. Similarly, the DEPARTMENT_ID column references the ID of a row in the DEPARTMENT_LOOKUP table and CLOTHING_ID references the ID of a row in the CLOTHING_LOOKUP table.

| Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries |
|--|------------|---------------|---------------|-------------|--------|------------|-------------|--|---------------------|------------|------|----------------|
| | Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table | Create App | | |
| Column Name | | | Data Type | | | Nullable | | Default | | | | |
| PRODUCT_ID | | | NUMBER | | | No | | "WKSP_APEXHANDSONLAB22"."ISEQ\$\$_173406532".nextval | | | | |
| PRODUCT_NAME | | | VARCHAR2(255) | | | No | | - | | | | |
| UNIT_PRICE | | | NUMBER(10,2) | | | Yes | | - | | | | |
| PRODUCT_DETAILS | | | BLOB | | | Yes | | - | | | | |
| PRODUCT_IMAGE | | | BLOB | | | Yes | | - | | | | |
| IMAGE_MIME_TYPE | | | VARCHAR2(512) | | | Yes | | - | | | | |
| IMAGE_FILENAME | | | VARCHAR2(512) | | | Yes | | - | | | | |
| IMAGE_CHARSET | | | VARCHAR2(512) | | | Yes | | - | | | | |
| IMAGE_LAST_UPDATED | | | DATE | | | Yes | | - | | | | |
| COLOR_ID | | | NUMBER | | | Yes | | - | | | | |
| DEPARTMENT_ID | | | NUMBER | | | Yes | | - | | | | |
| CLOTHING_ID | | | NUMBER | | | Yes | | - | | | | |
| Download Print | | | | | | | | | | | | |

You now know how to add new columns to your existing tables, how to create lookup tables for reference information, and how to create and run a SQL script to populate your tables. You may now **proceed to the next practice**.

Practice 3: Create a Database Package for Business Logic

Overview

In this practice, you learn to create database objects to use in your APEX application. This package contains functions and procedures to add products to the cart, remove products, create an order, clear the cart, and more.

To manage items in the cart, you use [collections](#), which enable you to temporarily store products currently in session state so they can be accessed, manipulated, or processed during a user's specific session.

Business logic in APEX applications can be written using PL/SQL, Oracle's procedural language extension to SQL. PL/SQL offers a handy program unit called a "package" that allows you to cleanly separate the API signatures for a set of reusable procedures and functions, from their implementation.

In this practice, you will:

- Create a package to manage the Shopping Cart

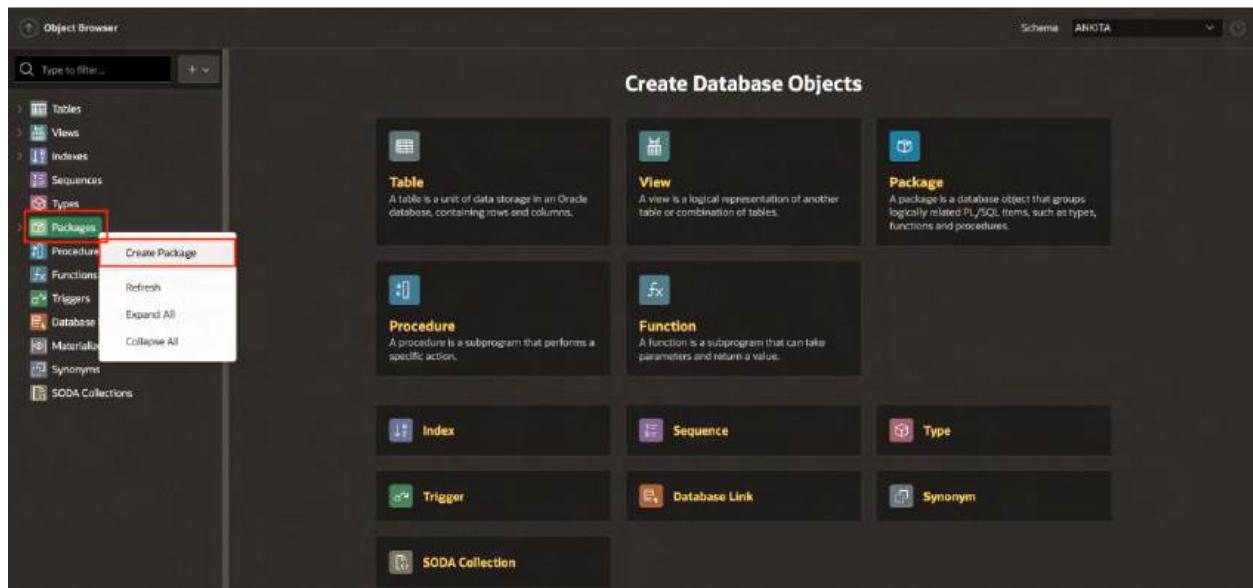
Task

Create the Package

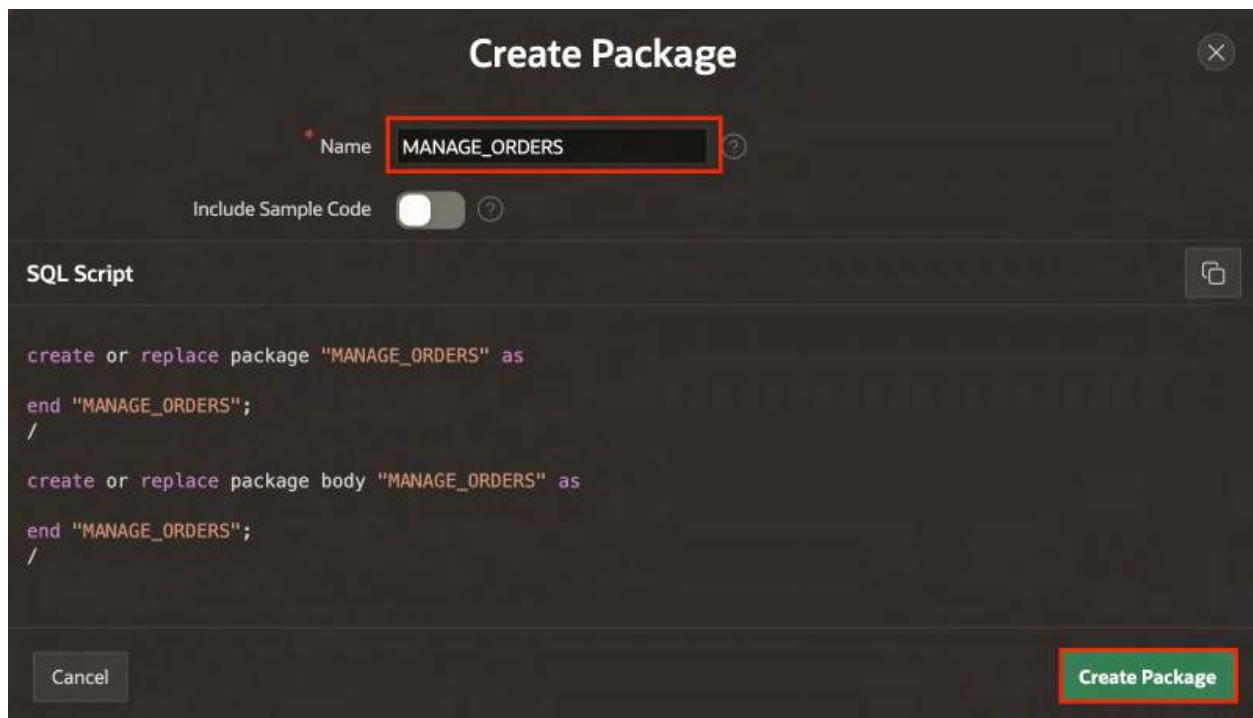
Create specification and body for the package.

1. Navigate to **SQL Workshop**, and click **Object Browser**.

2. Navigate to the **Packages** on the left side, Right click on it and Select **Create Package**.



3. For Package Name, enter **MANAGE_ORDERS** and click **Create Package**.



4. For Specification, replace the contents of the code editor with the following:

```
CREATE OR replace PACKAGE manage_orders
AS
-----
-- create procedure for add a product temporarily
PROCEDURE add_product (
    p_product IN NUMBER,
    p_quantity IN NUMBER);

-----
-- create procedure for remove a product temporarily
PROCEDURE remove_product (
    p_product IN NUMBER);

-----
-- create function to get the number of items in the shopping
cart
FUNCTION Get_quantity
RETURN NUMBER;

-----
-- create procedure for validate if a product exists in the
shopping cart
FUNCTION Product_exists(
    p_product IN NUMBER)
RETURN NUMBER;

-----
-- create procedure for clear the cart
PROCEDURE clear_cart;

-----
-- create function to validate a customer
FUNCTION Customer_exists(
    p_customer_email IN VARCHAR2)
RETURN NUMBER;

-----
-- create procedure to insert orders
PROCEDURE create_order (
    p_customer          IN VARCHAR2 DEFAULT NULL,
    p_customer_email   IN VARCHAR2,
    p_store             IN NUMBER,
    p_order_id          OUT orders.order_id%TYPE,
    p_customer_id       OUT NUMBER );
END manage_orders;
```

5. Click **Save and Compile** to save the changes.

```
CREATE OR REPLACE PACKAGE manage_orders
AS
  -- create procedure for add a product temporarily
  PROCEDURE add_product (
    p_product IN NUMBER,
    p_quantity IN NUMBER);

  -- create procedure for remove a product temporarily
  PROCEDURE remove_product (
    p_product IN NUMBER);

  -- create function to get the number of items in the shopping cart
  FUNCTION Get_quantity
  RETURN NUMBER;

  -- create procedure for validate if a product exists in the shopping cart
  FUNCTION Product_exists(
    p_product IN NUMBER)
  RETURN NUMBER;

  -- create procedure for clear the cart
  PROCEDURE clear_cart;

  -- create function to validate a customer
  FUNCTION Customer_exists(
    p_customer_email IN VARCHAR2)
  RETURN NUMBER;
END;
```

6. Navigate to the body part of the package by clicking the Body tab and replace the contents of the code editor with the following:

```
CREATE OR replace PACKAGE BODY manage_orders
AS
    PROCEDURE add_product (p_product    IN NUMBER,
                           p_quantity   IN NUMBER)
    IS
    BEGIN
        IF NOT apex_collection.collection_exists
(p_collection_name => 'PRODUCTS')
        THEN
            apex_collection.create_collection(p_collection_name =>
'PRODUCTS');
        END IF;

        apex_collection.add_member(p_collection_name =>
'PRODUCTS',
                           p_n001 => p_product,
                           p_n002 => p_quantity);
    END add_product;

    PROCEDURE remove_product (p_product IN NUMBER)
    IS
        l_id NUMBER;
    BEGIN
        IF apex_collection.Collection_exists (p_collection_name =>
'PRODUCTS')
        THEN
            SELECT seq_id
            INTO   l_id
            FROM   apex_collections a
            WHERE  collection_name = 'PRODUCTS'
                   AND a.n001 = p_product;

            apex_collection.delete_member(p_collection_name =>
'PRODUCTS',
                           p_seq => l_id);
        END IF;
    END remove_product;

    FUNCTION get_quantity
    RETURN NUMBER
    IS
```

```

    l_items NUMBER := 0;
BEGIN
    IF apex_collection.collection_exists (p_collection_name =>
'PRODUCTS')
    THEN
        SELECT SUM(n002)
        INTO l_items
        FROM apex_collections a
        WHERE collection_name = 'PRODUCTS';
    END IF;

    RETURN l_items;
END get_quantity;

FUNCTION product_exists(p_product IN NUMBER)
RETURN NUMBER
IS
    l_quantity NUMBER;
BEGIN
    IF apex_collection.collection_exists (p_collection_name =>
'PRODUCTS')
    THEN
        SELECT a.n002
        INTO l_quantity
        FROM apex_collections a
        WHERE collection_name = 'PRODUCTS'
            AND a.n001 = p_product;

        RETURN l_quantity;
    ELSE
        RETURN 0;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 0;
END product_exists;

PROCEDURE clear_cart
IS
BEGIN
    IF apex_collection.collection_exists (p_collection_name =>
'PRODUCTS')
    THEN

```

```

        apex_collection.truncate_collection(p_collection_name =>
'PRODUCTS');
    END IF;
END clear_cart;

FUNCTION customer_exists(p_customer_email IN VARCHAR2)
RETURN NUMBER
IS
    l_customer customers.customer_id%TYPE;
BEGIN
    SELECT customer_id
    INTO    l_customer
    FROM    customers
    WHERE   email_address = p_customer_email;

    RETURN l_customer;
EXCEPTION
    WHEN no_data_found THEN
        RETURN 0;
END customer_exists;

PROCEDURE create_order (p_customer           IN VARCHAR2,
                        p_customer_email IN VARCHAR2,
                        p_store          IN NUMBER,
                        p_order_id       OUT
orders.order_id%TYPE,
                        p_customer_id    OUT NUMBER)
IS
BEGIN
    p_customer_id := customer_exists(p_customer_email);

    IF p_customer_id = 0 THEN
        INSERT INTO customers
            (full_name,
             email_address)
        VALUES      (p_customer,
                     p_customer_email)
        returning customer_id INTO p_customer_id;
    END IF;

    INSERT INTO orders
        (order_datetime,
         customer_id,

```

```

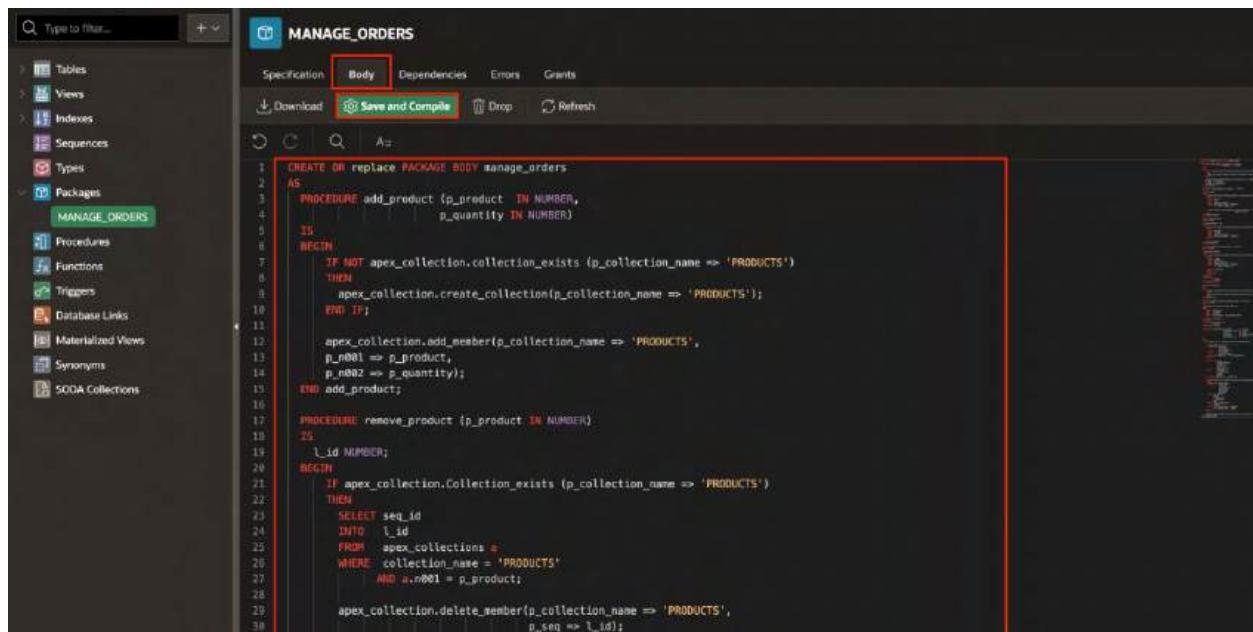
        store_id,
        order_status)
VALUES      (SYSDATE,
            p_customer_id,
            p_store,
            'OPEN')
    returning order_id INTO p_order_id;

    IF apex_collection.collection_exists (p_collection_name =>
'PRODUCTS')
    THEN
        INSERT INTO order_items
            (order_id,
            line_item_id,
            product_id,
            unit_price,
            quantity)
        SELECT p_order_id,
            seq_id,
            p.product_id,
            p.unit_price,
            n002
        FROM   apex_collections a,
            products p
        WHERE  collection_name = 'PRODUCTS'
            AND p.product_id = a.n001;
    END IF;

    apex_collection.delete_collection(p_collection_name =>
'PRODUCTS');
END create_order;
END manage_orders;

```

7. Click **Save & Compile**.



The screenshot shows the Oracle SQL Workshop interface. On the left, there's a sidebar with various database objects like Tables, Views, Indexes, Sequences, Packages, Procedures, Functions, Triggers, Database Links, Materialized Views, Synonyms, and SODA Collections. Under the 'Packages' section, 'MANAGE_ORDERS' is selected. The main area is titled 'MANAGE_ORDERS' and has tabs for Specification, Body, Dependencies, Errors, and Grants. The 'Body' tab is active, showing the PL/SQL code for the package. A red box highlights the 'Save and Compile' button at the top of the code editor. The code itself is a package named 'MANAGE_ORDERS' with two procedures: 'add_product' and 'remove_product'. The 'add_product' procedure checks if a collection exists for 'PRODUCTS', creates it if not, adds the product to the collection, and then inserts a new row into the 'CUSTOMERS' table with a generated sequence value. The 'remove_product' procedure deletes the product from the collection and updates the 'CUSTOMERS' table.

```
CREATE OR REPLACE PACKAGE BODY manage_orders AS
  PROCEDURE add_product (p_product IN NUMBER,
                        p_quantity IN NUMBER)
  IS
    BEGIN
      IF NOT apex_collection.collection_exists (p_collection_name => 'PRODUCTS')
      THEN
        apex_collection.create_collection(p_collection_name => 'PRODUCTS');
      END IF;

      apex_collection.add_member(p_collection_name => 'PRODUCTS',
                                p_nb01 => p_product,
                                p_nb02 => p_quantity);
    END add_product;

  PROCEDURE remove_product (p_product IN NUMBER)
  IS
    l_id NUMBER;
  BEGIN
    IF apex_collection.collection_exists (p_collection_name => 'PRODUCTS')
    THEN
      SELECT seq_id
      INTO l_id
      FROM apex_collections a
      WHERE collection_name = 'PRODUCTS'
        AND a.n001 = p_product;

      apex_collection.delete_member(p_collection_name => 'PRODUCTS',
                                    p_seq => l_id);
    END IF;
  END remove_product;
END;
```

While you don't have to understand the code to complete the workshop successfully, know that the functions and procedures you've defined in this practice are using a built-in feature of Oracle APEX to handle the user's shopping cart by managing a collection of product id and quantity values specific to the current user, and automatically create a new row in the **Customers** table during order creation if it's the first time the user is placing an order.

You now know how to create a package to manage the shopping cart. In the following practices, you will call these procedures and functions when it is required. You may now **proceed to the next practice**.

Practice 4: Create Database Objects Using Quick SQL

Overview

Memorizing the SQL to create and maintain database objects can be difficult. However, there are tools that can generate the code for you based on a shorthand syntax or a graphical user interface. In this practice, you will create database objects using Quick SQL and then view the objects in Database Actions. Then you will then use the Data Workshop page to load data from a file into your workspace.

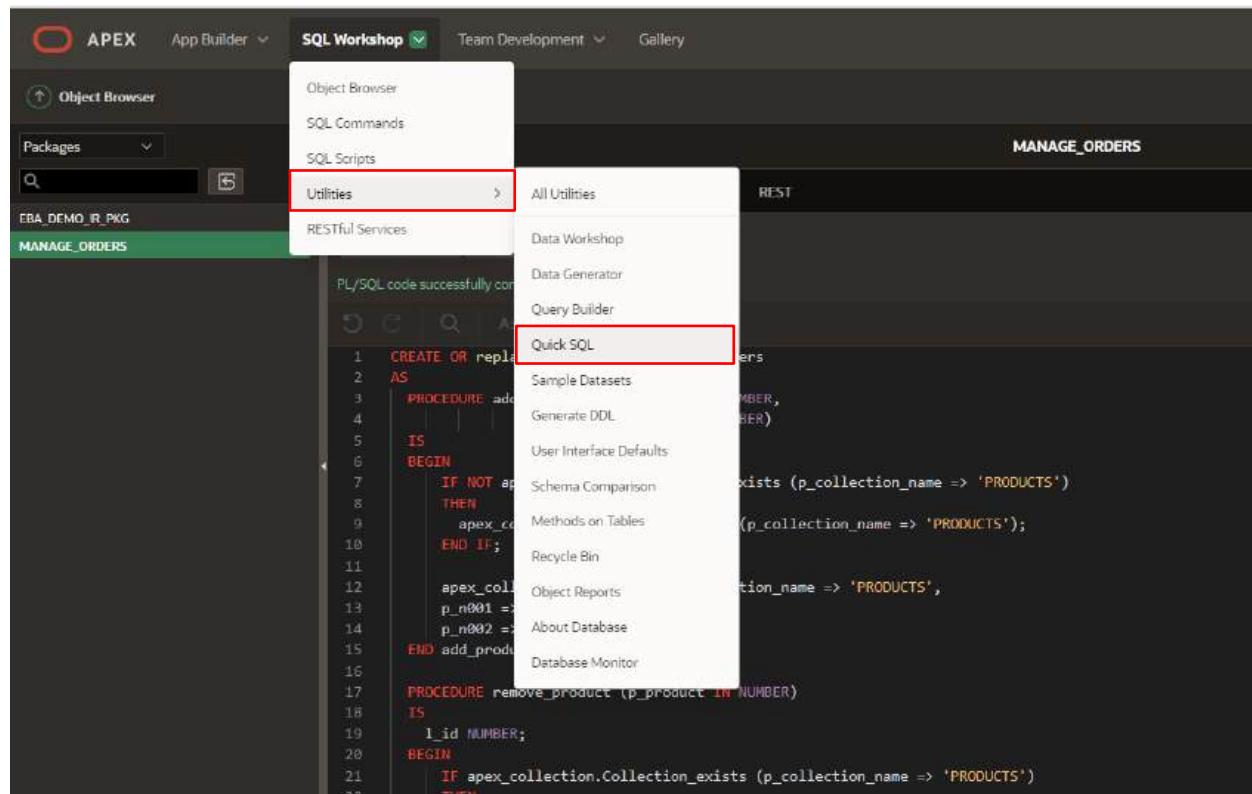
In this practice, you'll:

- Create database objects using Quick SQL
- Load data using the Data Workshop page

Tasks

Create database objects using Quick SQL

1. Return to your APEX Workspace and click 'v' next to the **SQL Workshop** tab, click **Utilities**, and then select **Quick SQL**.



2. Copy the following SQL shorthand code and paste it into the left pane in Quick SQL.

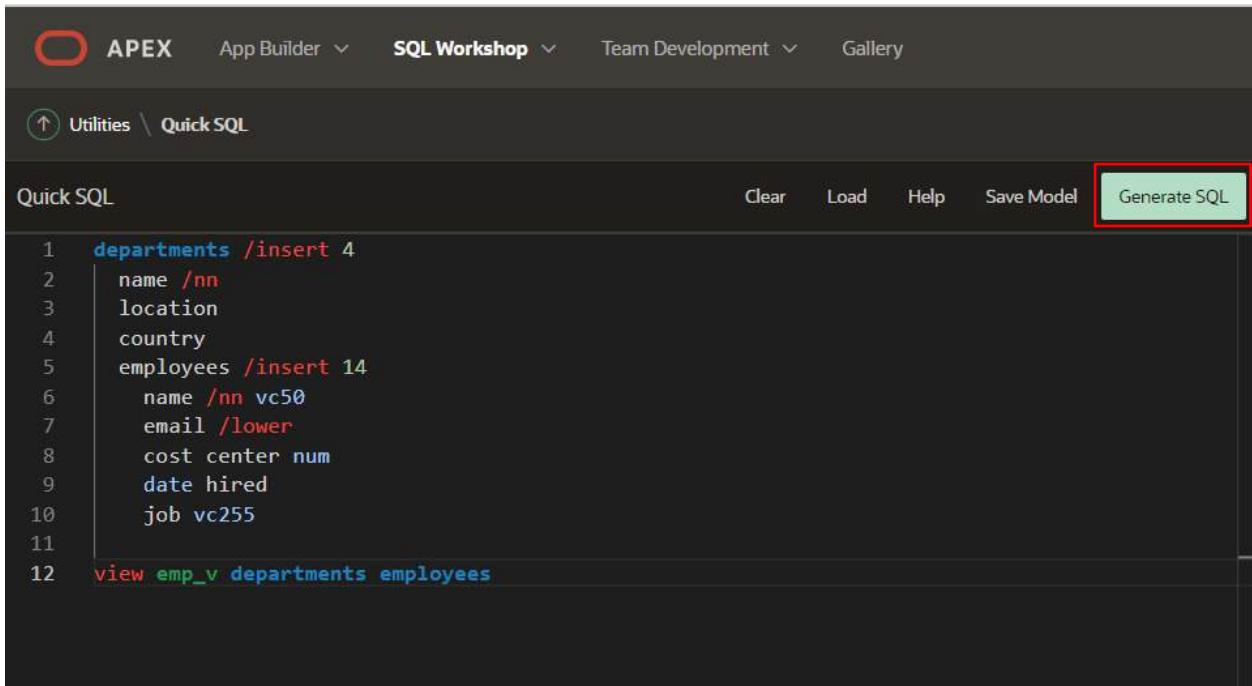
```
departments /insert 4
  name /nn
  location
  country
employees /insert 14
  name /nn vc50
  email /lower
  cost center num
  date hired
  job vc255

view emp_v departments employees
```

Note that the Indentation from the code you copied should be similar to the one we have in the following screenshot.

3. Review the shorthand code. How many tables will be created and how are they related? What types of columns and constraints will be created for the tables? Note that the **Help** button provides details on the shorthand syntax along with several examples of its usage.

When ready, in the Quick SQL (left pane) toolbar, click **Generate SQL**.



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. Below the navigation is a breadcrumb trail Utilities \ Quick SQL. The main area is titled "Quick SQL". On the right side of the toolbar are buttons for Clear, Load, Help, Save Model, and Generate SQL, with the "Generate SQL" button highlighted by a red rectangle. The left pane contains the SQL shorthand code provided in the task, with line numbers 1 through 12 on the left. The code defines two tables: "departments" with 4 rows and "employees" with 14 rows, and a view "emp_v" that joins "departments" and "employees".

```
1  departments /insert 4
2  name /nn
3  location
4  country
5  employees /insert 14
6    name /nn vc50
7    email /lower
8    cost center num
9    date hired
10   job vc255
11
12  view emp_v departments employees
```

4. Review the SQL that is generated in the right-hand pane, and then click **Settings** to make some changes to the resulting code.

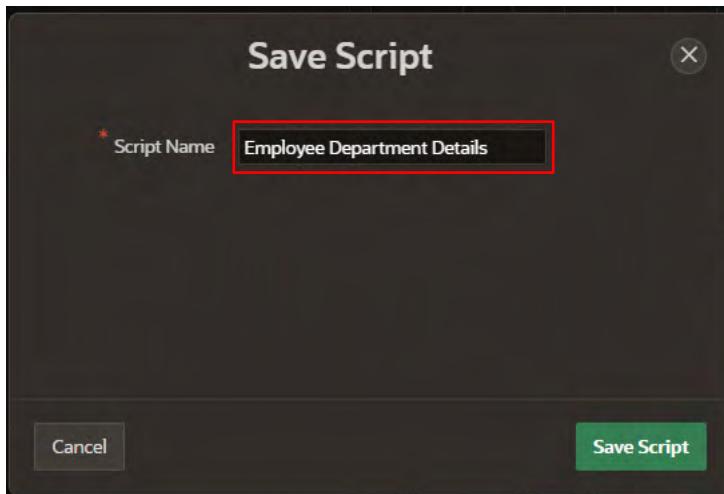
It is important to save the SQL Shorthand as a script so that it can be run in your database. In the SQL (right pane) toolbar, click **Save SQL Script**.

The screenshot shows the Oracle SQL Workshop interface. The left pane contains SQL shorthand code for creating departments and employees. The right pane shows the generated SQL script. The 'Save SQL Script' button in the top right corner is highlighted with a red box.

```
1 departments /insert 4
2   name /m
3   location
4   country
5 employees /insert 14
6   name /m vc50
7   email /lower
8   cost_center num
9   date_hired
10  job vc255
11
12 view emp_v departments employees
```

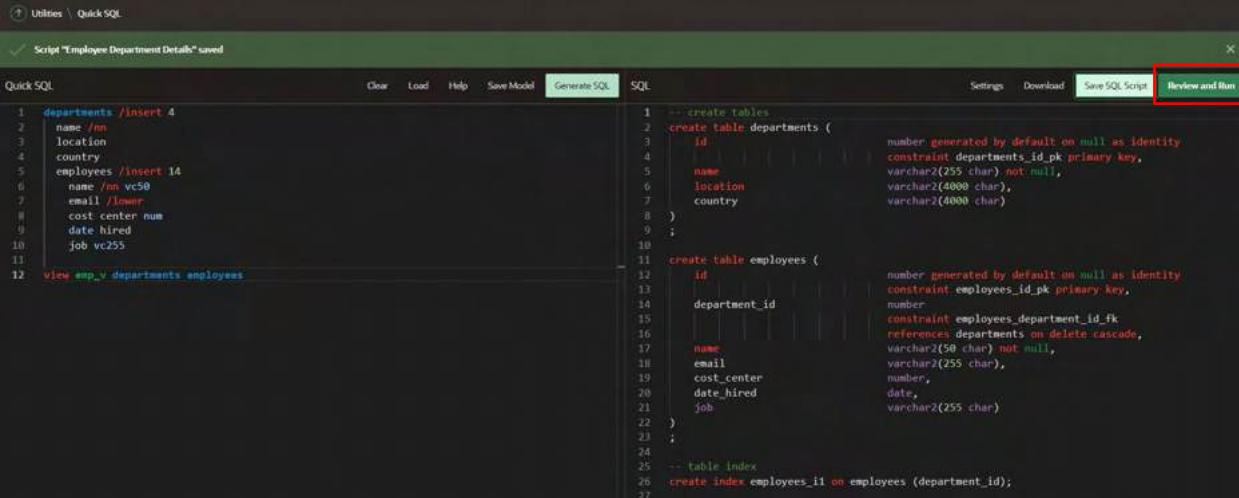
```
1  -- create tables
2  create table departments (
3    id
4    name
5    location
6    country
7  );
8
9;
10
11 create table employees (
12   id
13   department_id
14   number
15   constraint employees_id_pk primary key,
16   constraint employees_department_id_fk
17   references departments on delete cascade,
18   name
19   email
20   cost_center
21   date_hired
22 );
23
24
25 -- table index
26 create index employees_id_i on employees (department_id);
27
28
29
```

5. In the Save Script dialog, for Script Name enter **Employee Department Details**, and click **Save Script**.



6. Now that you saved the script you can run it to create the specified database objects.

In the SQL (right pane) toolbar, click **Review and Run**.



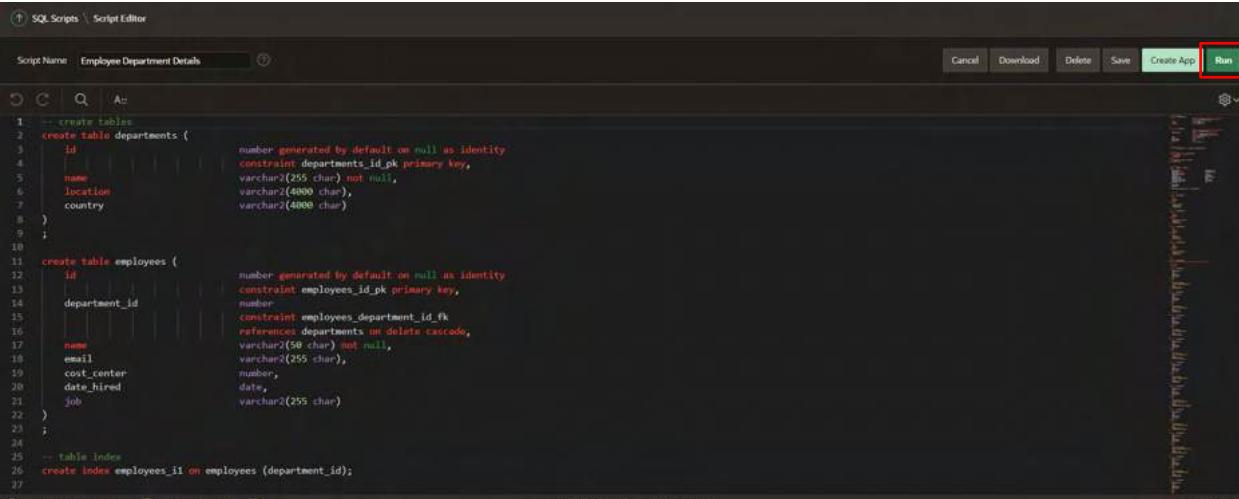
The screenshot shows the Oracle SQL Workshop interface. In the top right corner of the main window, there is a toolbar with several buttons. One of these buttons, labeled "Review and Run", is highlighted with a red box. The central area of the window displays two blocks of SQL code. The left block contains DML statements for inserting data into tables named "departments" and "employees". The right block contains DDL statements for creating the "departments" and "employees" tables, including their columns and constraints like primary keys and foreign keys. Both blocks end with a comment "-- table index" followed by a create index statement.

```
departments /insert 4
name /m
location
country
employees /insert 14
name /m v50
email /lower
cost_center num
date_hired
job v255
view emp_v departments employees

-- create tables
create table departments (
    id number generated by default on null as identity
        constraint departments_id_pk primary key,
    name varchar2(255 char) not null,
    location varchar2(4000 char),
    country varchar2(4000 char)
);

create table employees (
    id number generated by default on null as identity
        constraint employees_id_pk primary key,
    department_id number
        constraint employees_department_id_fk
            references departments on delete cascade,
    name varchar2(50 char) not null,
    email varchar2(255 char),
    cost_center number,
    date_hired date,
    job varchar2(255 char)
);
-- table index
create index employees_ix on employees (department_id);
```

7. In the Script Editor page toolbar, click **Run**.

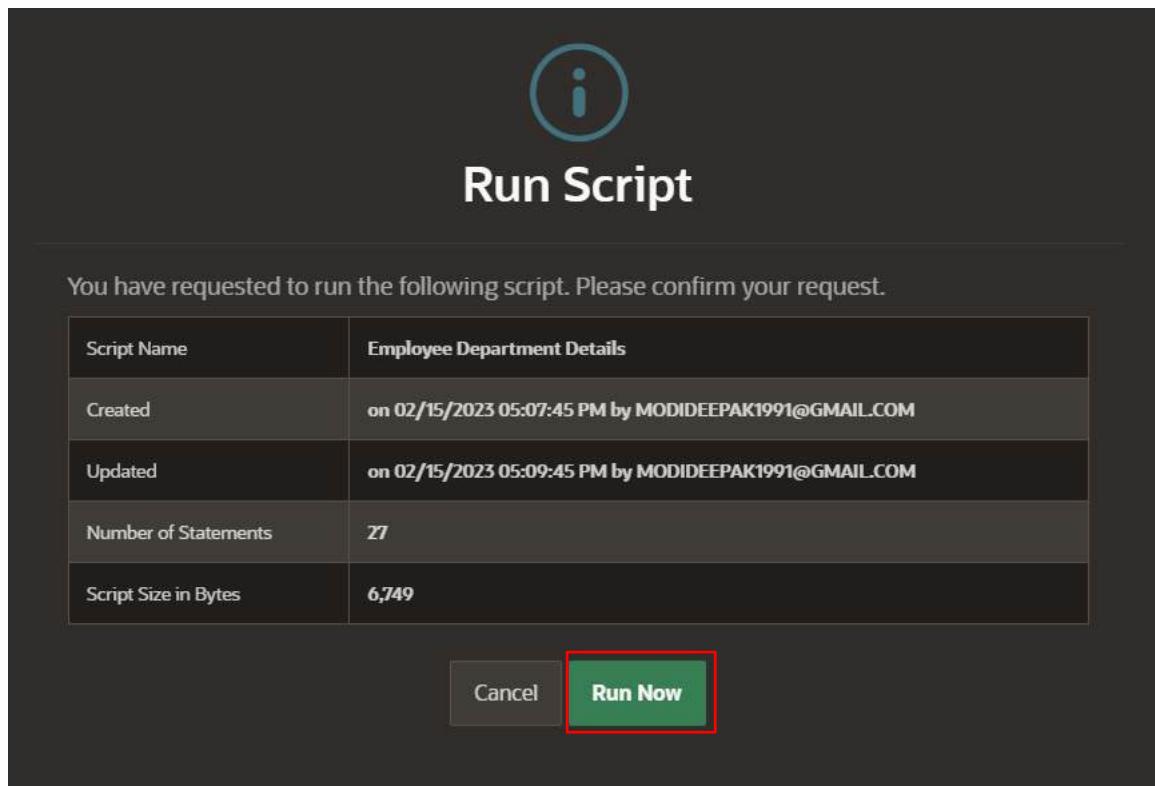


The screenshot shows the Oracle SQL Workshop Script Editor. At the top right of the editor window, there is a toolbar with several buttons. One of these buttons, labeled "Run", is highlighted with a red box. The main area of the editor contains the same two blocks of SQL code as the previous screenshot: DML for inserting data into "departments" and "employees", and DDL for creating the tables with their respective columns and constraints. The code ends with a create index statement for the "employees" table.

```
-- create tables
create table departments (
    id number generated by default on null as identity
        constraint departments_id_pk primary key,
    name varchar2(255 char) not null,
    location varchar2(4000 char),
    country varchar2(4000 char)
);

create table employees (
    id number generated by default on null as identity
        constraint employees_id_pk primary key,
    department_id number
        constraint employees_department_id_fk
            references departments on delete cascade,
    name varchar2(50 char) not null,
    email varchar2(255 char),
    cost_center number,
    date_hired date,
    job varchar2(255 char)
);
-- table index
create index employees_ix on employees (department_id);
```

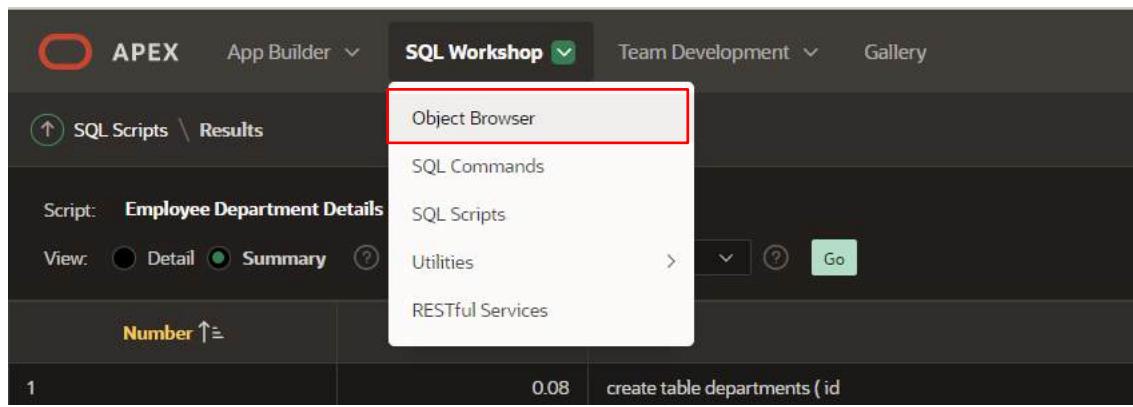
8. On the Run Script page, click **Run Now**.



9. The Results page shows the results of running the script. Scroll to the bottom to see a summary. You should not see any errors.

| SQL Scripts \ Results | | | | | |
|-------------------------------------|---------|--|----------------------|------|--|
| Script: Employee Department Details | | Status: Complete | | | |
| View: | Detail | Summary | Rows: 15 | Go | |
| | | | | | |
| Number ↑↓ | Elapsed | Statement | Feedback | Rows | |
| 1 | 0.08 | create table departments (id | Table created. | 0 | |
| 2 | 0.05 | create table employees (id | Table created. | 0 | |
| 3 | 0.02 | create index employees_ix on employees (department_id) | Index created. | 0 | |
| 4 | 0.04 | create or replace trigger employees_biu before insert or | Trigger created. | 0 | |
| 5 | 0.06 | create or replace view emp_v as select departments.id | View created. | 0 | |
| 6 | 0.02 | insert into departments (id, name, location, | 1 row(s) inserted. | 1 | |
| 7 | 0.01 | insert into departments (id, name, location, | 1 row(s) inserted. | 1 | |
| 8 | 0.00 | insert into departments (id, name, location, | 1 row(s) inserted. | 1 | |
| 9 | 0.00 | insert into departments (id, name, location, | 1 row(s) inserted. | 1 | |
| 10 | 0.00 | commit | Statement processed. | 0 | |
| 11 | 0.07 | alter table departments modify id generated always as identity | Table altered. | 0 | |
| 12 | 0.08 | insert into employees (id, department_id, name, | 1 row(s) inserted. | 1 | |
| 13 | 0.00 | insert into employees (id, department_id, name, | 1 row(s) inserted. | 1 | |
| 14 | 0.00 | insert into employees (id, department_id, name, | 1 row(s) inserted. | 1 | |
| 15 | 0.00 | insert into employees (id, department_id, name, | 1 row(s) inserted. | 1 | |

10. To view the database objects that were created, click the arrow 'v' in the **SQL Workshop** tab and then select **Object Browser**.



11. Select the **EMPLOYEES** table and review its columns.

The screenshot shows the Oracle APEX Object Browser interface. On the left, a tree view lists various database objects, with 'EMPLOYEES' selected and highlighted with a red box. The main panel displays the 'EMPLOYEES' table structure with the following columns:

| Column Name | Data Type | Nullable | Default | Primary Key | Comment | Identity |
|---------------|--------------------|----------|-------------------------------------|-------------|---------|------------|
| ID | NUMBER | N | 'ANIKITA_BER'.'SEQ\$\$_18825292...' | 1 | | BV DEFAULT |
| DEPARTMENT_ID | NUMBER | Y | | | | |
| NAME | VARCHAR2(50 CHAR) | N | | | | |
| EMAIL | VARCHAR2(255 CHAR) | Y | | | | |
| COST_CENTER | NUMBER | Y | | | | |
| DATE_HIRED | DATE | Y | | | | |
| JOB | VARCHAR2(255 CHAR) | Y | | | | |

- Click the **Data** tab to see the data that was loaded.

The screenshot shows the Object Browser on the left with various schema objects listed. The main area displays the **EMPLOYEES** table with the following columns: ID, DEPARTMENT_ID, NAME, EMAIL, COST_CENTER, DATE_HIRED, and JOB. The data is as follows:

| ID | DEPARTMENT_ID | NAME | EMAIL | COST_CENTER | DATE_HIRED | JOB |
|----|---------------|--------------------|------------------------------------|-------------|------------|------------------------------|
| 1 | 5 | Gricelda Luebbers | gricelda.luebbers@aa...@aaag.com | 40 | 02/26/2023 | Manufacturing and Distrib... |
| 2 | 2 | Dean Bollich | dean.bollich@a...@aa.com | 14 | 03/27/2023 | Java Developer |
| 3 | 1 | Milo Manoni | milo.manoni@aa...@aa.com | 59 | 02/25/2023 | Help Desk Specialist |
| 4 | 1 | Laurice Karl | laurice.karl@aa...@aa.com | 65 | 04/24/2023 | Systems Software Engineer |
| 5 | 1 | August RupeI | august.rupeI@aa...@aa.com | 91 | 03/26/2023 | Program Manager |
| 6 | 1 | Salome Gultti | salome.gultti@aa...@aaag.com | 14 | 02/27/2023 | Programmer Analyst |
| 7 | 1 | Lovie Rtacco | lovie.rtacco@aa...@aa.com | 53 | 02/19/2023 | Analyst |
| 8 | 4 | Chaya Graczkowski | chaya.graczkowsk...@aa...@aaag.com | 98 | 02/24/2023 | Accounting Analyst |
| 9 | 1 | Twila Coolbeth | twila.coolbeth@aa...@aa.com | 32 | 03/28/2023 | Quality Assurance Analyst |
| 10 | 3 | Carlotta Achenbach | carlotta.achenbach@aa...@aa.com | 54 | 04/10/2023 | Java Developer |
| 11 | 3 | Jeraldine Audet | jeraldine.audet@aa...@aa.com | 6 | 03/02/2023 | Application Developer |
| 12 | 1 | August Arouri | august.arouri@aa...@aa.com | 71 | 02/07/2023 | Application Developer |
| 13 | 4 | Ward Stepney | ward.stepney@aa...@aa.com | 13 | 04/02/2023 | HR Representative |
| 14 | 2 | Ayana Berkhurst | ayana.berkhurst@aa...@aa.com | 42 | 01/29/2023 | Project Manager |

Load Data Using the Data Workshop Utility

In this step, you will learn how to create a table and Load data using the Data Workshop utility.

- On the Workspace home page, click the '**v**' next to **SQL Workshop** tab, click **Utilities**, and then select **Data Workshop**.

The screenshot shows the Object Browser on the left with various schema objects listed. The main area displays the **EMPLOYEES** table with the following columns: NAME, EMAIL, COST_CENTER, DATE_HIRED, and JOB. The data is as follows:

| NAME | EMAIL | COST_CENTER | DATE_HIRED | JOB |
|--------------------|------------------------------------|-------------|------------|------------------------------|
| Gricelda Luebbers | gricelda.luebbers@aa...@aaag.com | 40 | 02/26/2023 | Manufacturing and Distrib... |
| Dean Bollich | dean.bollich@a...@aa.com | 14 | 03/27/2023 | Java Developer |
| Milo Manoni | milo.manoni@aa...@aa.com | 59 | 02/25/2023 | Help Desk Specialist |
| Laurice Karl | laurice.karl@aa...@aa.com | 65 | 04/24/2023 | Systems Software Engineer |
| August RupeI | august.rupeI@aa...@aa.com | 91 | 03/26/2023 | Program Manager |
| Salome Gultti | salome.gultti@aa...@aaag.com | 14 | 02/27/2023 | Programmer Analyst |
| Lovie Rtacco | lovie.rtacco@aa...@aa.com | 53 | 02/19/2023 | Analyst |
| Chaya Graczkowski | chaya.graczkowsk...@aa...@aaag.com | 98 | 02/24/2023 | Accounting Analyst |
| Twila Coolbeth | twila.coolbeth@aa...@aa.com | 32 | 03/28/2023 | Quality Assurance Analyst |
| Carlotta Achenbach | carlotta.achenbach@aa...@aa.com | 54 | 04/10/2023 | Java Developer |
| Jeraldine Audet | jeraldine.audet@aa...@aa.com | 6 | 03/02/2023 | Application Developer |

2. Under Get Started, click **Load Data**. The Load Data Wizard appears.

The screenshot shows the Oracle APEX interface with the 'SQL Workshop' tab selected. The main title is 'Get Started'. Below it are two buttons: 'Load Data' (green background) and 'Unload Data' (grey background). A section titled 'Recently Created Tables' lists several tables with their last modified times: DEPARTMENTS, EMPLOYEES, CLOTHING_LOOKUP, DEPARTMENT_LOOKUP, COLOR_LOOKUP, ORDERS, and ORDER_ITEMS. All entries show a modification time of '6 minutes ago' except for CLOTHING_LOOKUP which was modified '105 minutes ago'.

3. Under Upload a File, click **Choose File**.

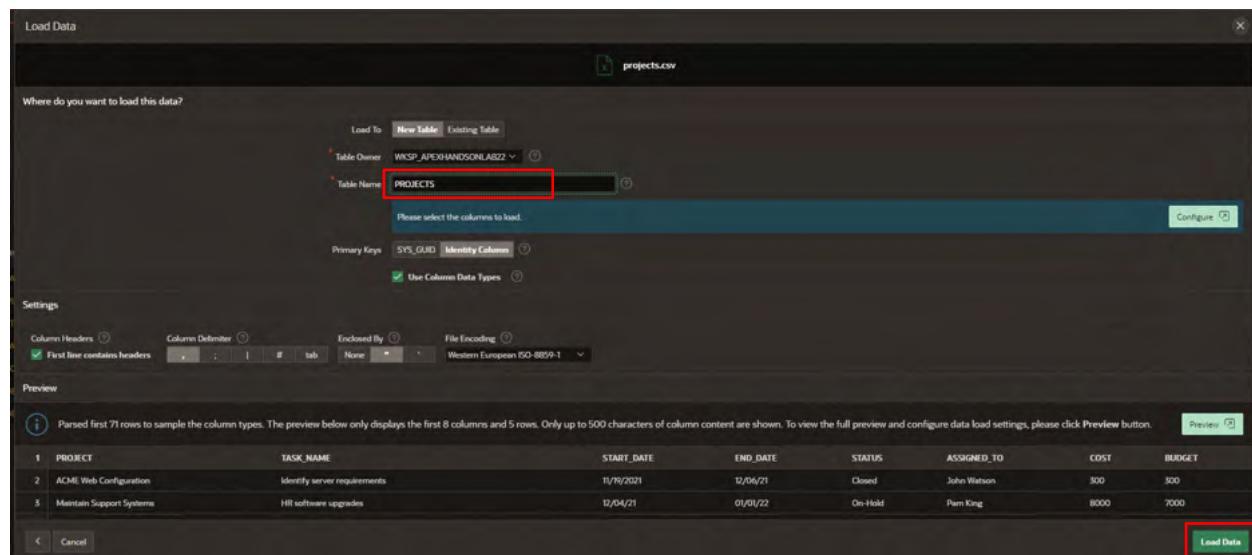
The screenshot shows the 'Load Data' wizard window. At the top, there are tabs for 'Upload a File' and 'Copy and Paste'. In the center, there is a large dashed rectangular area with the text 'Drag and drop file here or' above a green 'Choose File' button. Below the button, it says 'Supported formats csv, xlsx, txt, xml, json'.

4. Download [projects.csv](#) and select the file.

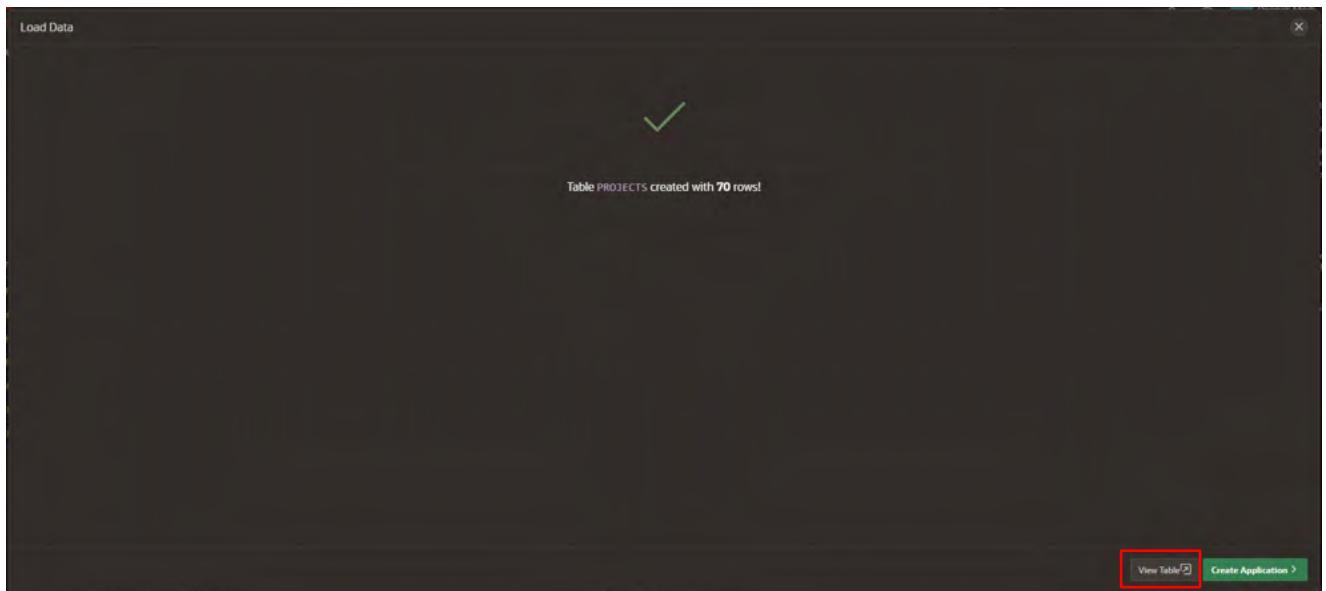
| | | | |
|---|--------------------|------------------------|----------|
| Lab 1 - Getting Started with Oracle APEX on ... | 2/15/2023 5:14 PM | Microsoft Word Doc... | 3,721 KB |
| Lab 2 - Use SQL Workshop.docx | 2/15/2023 9:27 PM | Microsoft Word Doc... | 3,665 KB |
| Lab 3 - Creating a Database Application.docx | 2/14/2023 5:33 PM | Microsoft Word Doc... | 220 KB |
| projects.csv | 2/15/2023 10:48 PM | Microsoft Excel Com... | 7 KB |
| Template.docx | 2/14/2023 2:55 PM | Microsoft Word Doc... | 251 KB |

Note: If the uploaded XLSX file contains multiple worksheets, the Load Data wizard picks the first sheet by default. To load another sheet, pick it from the Select Sheet select list. The First lines contains headers checkbox works similar to CSV files. The same is true for the Configure dialog, the behavior is the same as it is for CSV files. Uploading XLSX files is limited to 20 MB for each file.

5. Under **Load Data**, edit the Table Name field and provide an appropriate name. In this practice, you will set the Table Name as **Projects**. Then click **Load Data**.



6. Once Data Loading is finished, a success message is displayed. If some rows cannot be loaded to the target table, they will be stored to the error table and error table is displayed. Then click **View Table**.



7. Select the **PROJECTS** table and review its columns.

The screenshot shows the Oracle SQL Workshop Object Browser. On the left, under 'Tables', the 'PROJECTS' table is selected and highlighted in green. The main pane displays the 'PROJECTS' table structure with the following columns:

| Column Name | Data Type | Nullable | Default | Primary Key | Comment | Identity |
|-------------|--------------------|----------|--------------------------------|-------------|---------|------------|
| ID | NUMBER | N | 'ANKITA_BERI' ISQSS_1882529... | 1 | | BY DEFAULT |
| PROJECT | VARCHAR2(50 BYTE) | Y | | | | |
| TASK_NAME | VARCHAR2(255 BYTE) | Y | | | | |
| START_DATE | DATE | Y | | | | |
| END_DATE | DATE | Y | | | | |
| STATUS | VARCHAR2(50 BYTE) | Y | | | | |
| ASSIGNED_TO | VARCHAR2(50 BYTE) | Y | | | | |
| COST | NUMBER | Y | | | | |
| BUDGET | NUMBER | Y | | | | |

- Click the **Data** tab to see the data that was loaded.

The screenshot shows the Oracle SQL Workshop interface. On the left, the Object Browser lists various database objects: CLOTHING_LOOKUP, COLOR_LOOKUP, CUSTOMERS, DEMO_PROJECTS, DEPARTMENTS, DEPARTMENT_LOOKUP, EBA_DEMO_IR_DEPT, EBA_DEMO_IR_EMP, EBA_DEMO_IR_PROJECTS, EMPLOYEES, ORDERS, ORDER_ITEMS, PRODUCTS, PRODUCT_NAME_LOOKUP, and PROJECTS. The PROJECTS object is selected. The main pane displays the PROJECTS table with the following data:

| | ID | PROJECT | TASK_NAME | START_DATE | END_DATE | STATUS | ASSIGNED_TO | COST | BUDGET |
|----|----|--------------------|-----------------------|------------|------------|---------|---------------|------|--------|
| 1 | 1 | ACME Web Con... | Identify server r... | 11/19/2021 | 12/05/2021 | Closed | John Watson | 500 | 500 |
| 2 | 2 | Maintain Suppor... | HR software upg... | 12/04/2021 | 01/01/2022 | On-Hold | Pam King | 8000 | 7000 |
| 3 | 3 | Maintain Suppor... | Apply Billing Sys... | 01/07/2022 | 01/29/2022 | On-Hold | Russ Sanders | 9500 | 7000 |
| 4 | 4 | ACME Web Con... | Determine Web I... | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| 5 | 5 | ACME Web Con... | Specify security ... | 01/02/2022 | 01/04/2022 | Closed | John Watson | 200 | 300 |
| 6 | 6 | ACME Web Con... | Select servers fo... | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| 7 | 7 | Email Integration | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3000 | 1500 |
| 8 | 8 | ACME Web Con... | Configure Works... | 12/20/2021 | 01/04/2022 | Closed | John Watson | 200 | 100 |
| 9 | 9 | ACME Web Con... | Create pilot wor... | 01/15/2022 | 02/08/2022 | Closed | John Watson | 100 | 100 |
| 10 | 10 | ACME Web Con... | Run installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| 11 | 11 | Bug Tracker | Implement bug t... | 01/05/2022 | 01/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| 12 | 12 | Bug Tracker | Review automati... | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff | 2750 | 1500 |
| 13 | 13 | Train Developers | Publish develop... | 12/03/2021 | 12/13/2021 | On-Hold | John Watson | 1000 | 2000 |
| 14 | 14 | Train Developers | Publish links to s... | 12/28/2021 | 01/03/2022 | Closed | John Watson | 100 | 100 |

This completes the practice. At this point, you know how to use Quick SQL to generate database objects using a SQL shorthand. You also know how to use Data Workshop utility to load data.

Practice: Creating a Database Application

Get Started

Overview

In this lab, you will create a database application based on existing tables. You will also learn how to create an APEX application using data imported from a spreadsheet.

In this lab, you'll:

1. Create an application using the tables and data that you already have installed
2. Create a database application from a file

Practice 1: Create an Application Based on Existing Tables

Overview

In this practice, you will create a database application based on existing tables.

In this lab, you will:

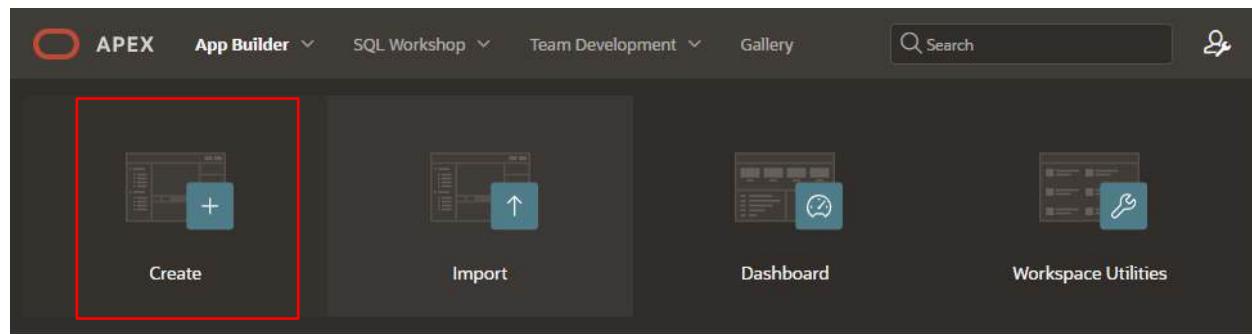
- Create an application using the tables and data that you already have installed
- Create a database application from a file

Tasks

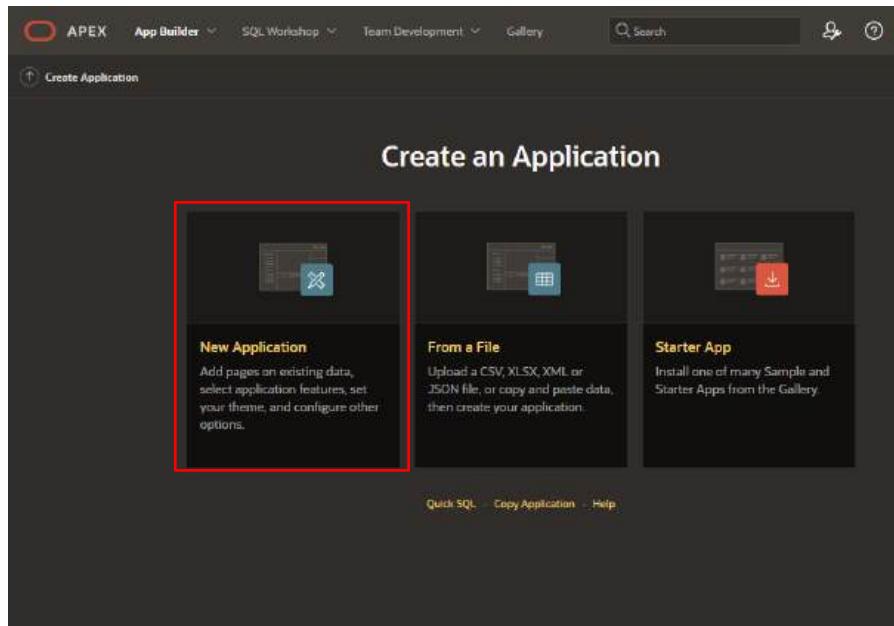
Sign in to the Oracle APEX Console using your workspace, username, and password.

Create an Application Based on Existing Tables

1. In the App Builder menu, click **App Builder**.
2. Click **Create**.

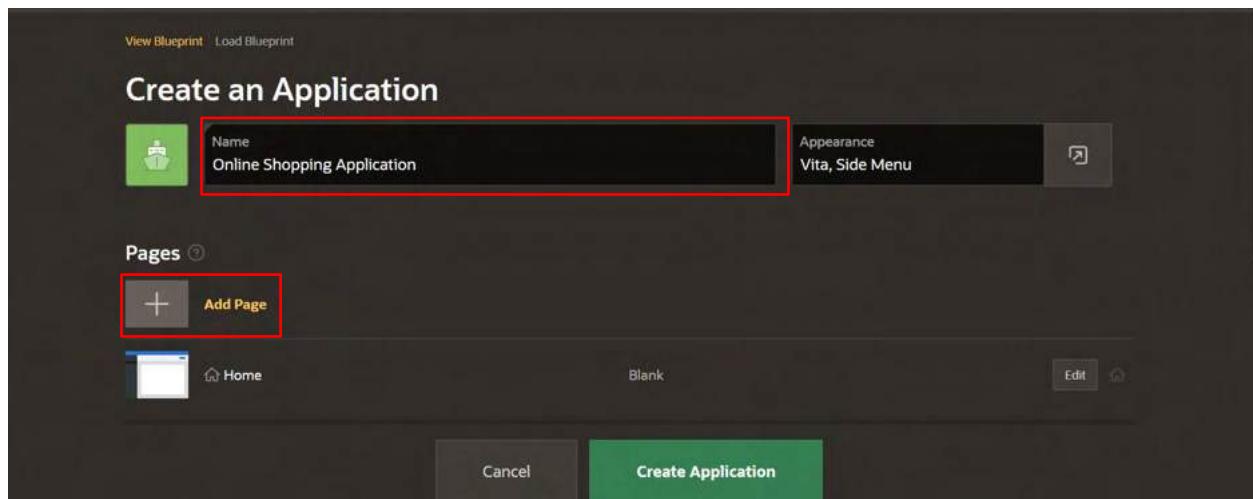


3. Click **New Application**.



Name the Application

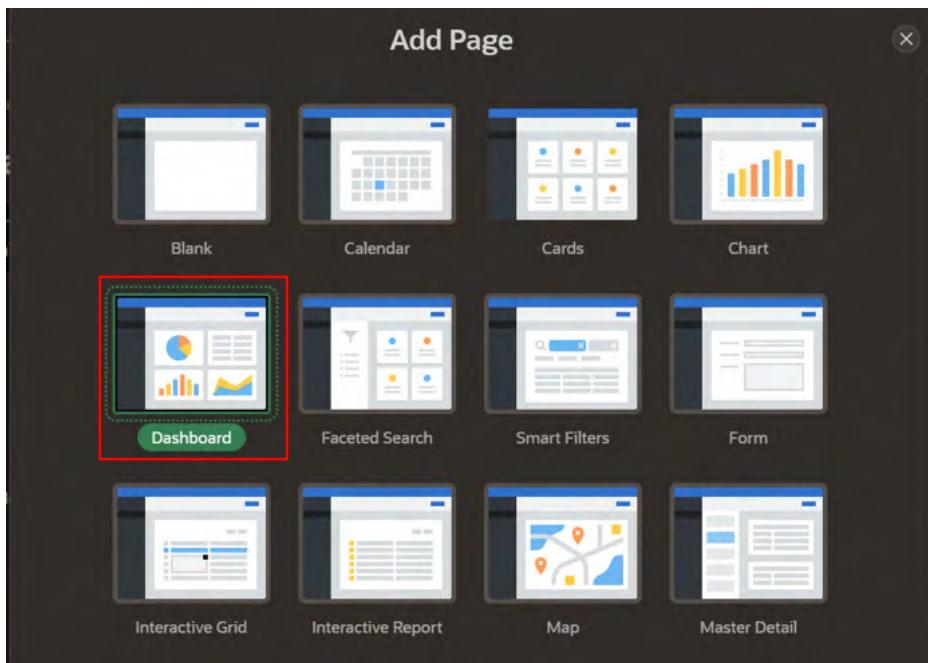
1. In the Create Application wizard, for “Name,” enter **Online Shopping Application**. Note that in the Create an Application wizard, the icon and color of the icon will be a random selection, so it most likely will not be the same color or icon that you see in the screenshot.



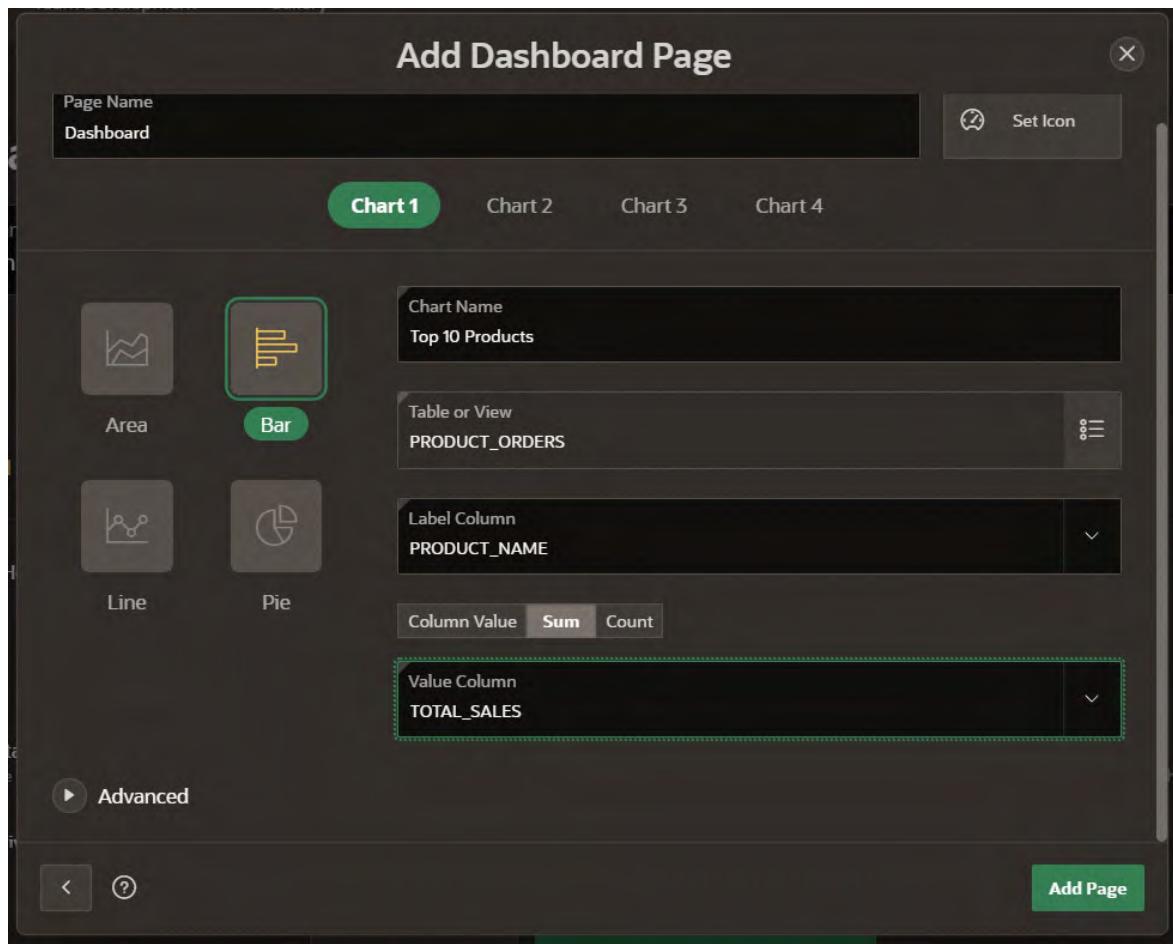
Add the Dashboard Page

A Dashboard page is a great way to show important information using various charts. When you installed the Sample Dataset, it also created a number of views, which join data from various tables. These views are ideal as the basis for the dashboard charts.

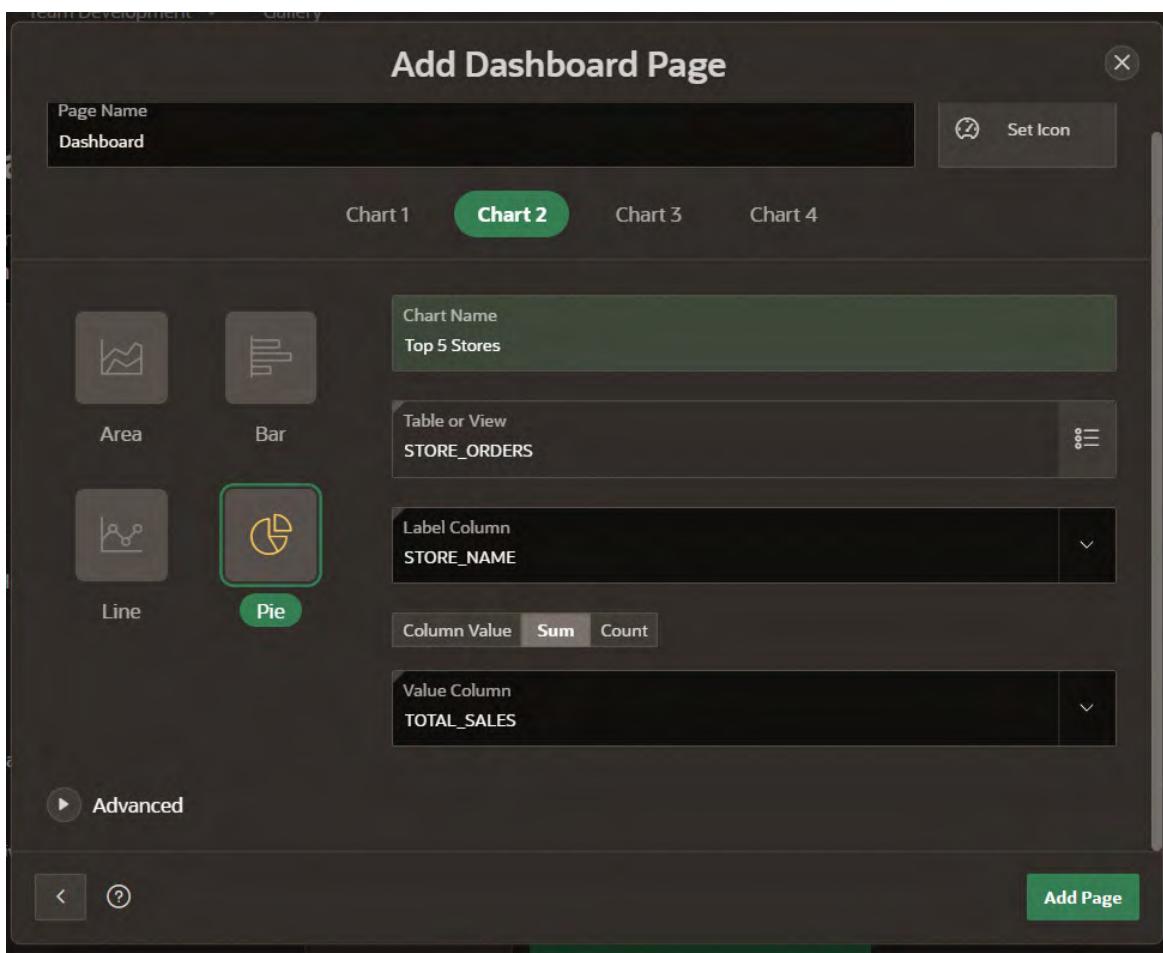
1. In the Create Application wizard, click **Add Page**.
2. Click **Dashboard**.



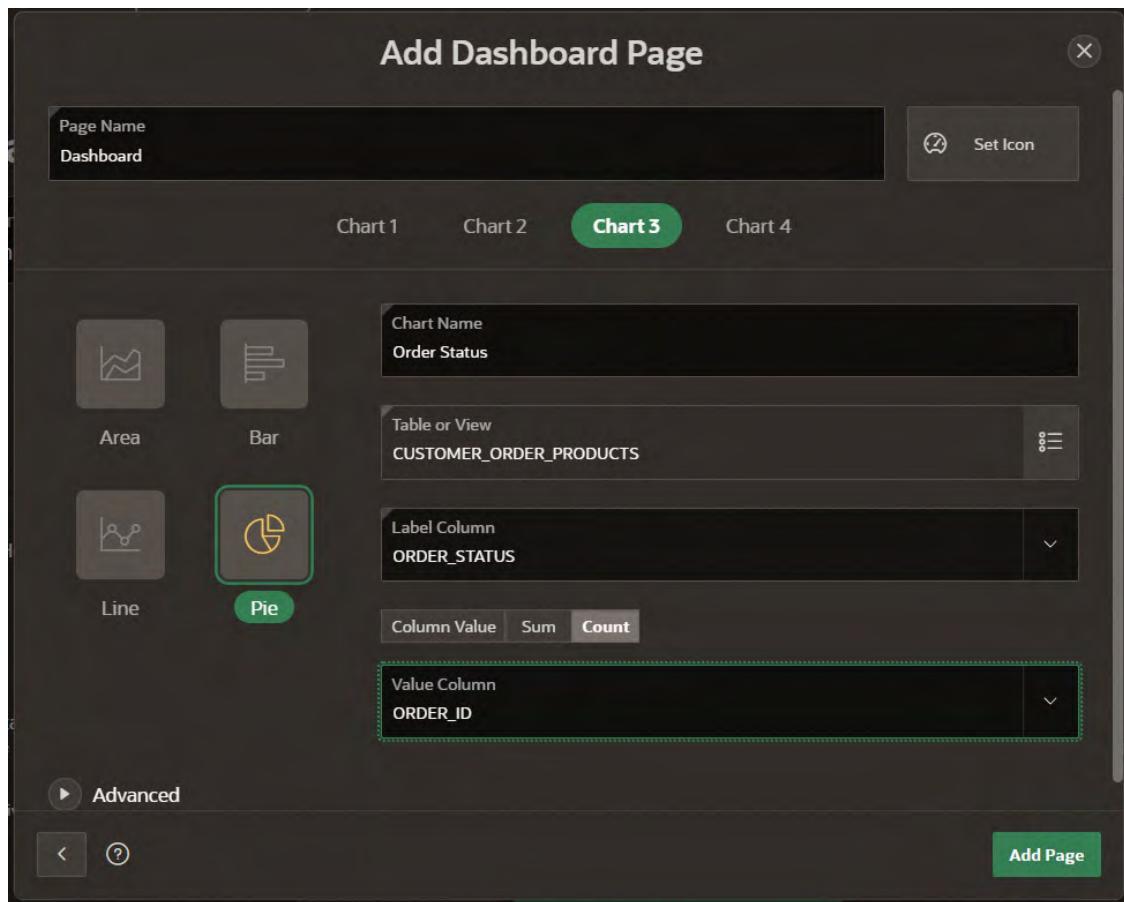
3. For Chart 1, enter the following:
 - Chart Type – select **Bar**
 - Chart Name – enter **Top 10 Products**
 - Table or View – select **PRODUCT_ORDERS**
 - Label Column – select **PRODUCT_NAME**
 - Type – select **Sum**
 - Value Column – select **TOTAL_SALES**



4. Click Chart 2, and enter the following:
 - Chart Type – select **Pie**
 - Chart Name – enter **Top 5 Stores**
 - Table or View – select **STORE_ORDERS**
 - Label Column – select **STORE_NAME**
 - Type – select **Sum**
 - Value Column – select **TOTAL_SALES**.



5. Click Chart 3, and enter the following:
 - Chart Type – select **Pie**
 - Chart Name – enter **Order Status**
 - Table or View – select **CUSTOMER_ORDER_PRODUCTS**
 - Label Column – select **ORDER_STATUS**
 - Type – select **Count**
 - Value Column – select **ORDER_ID**



6. Click Chart 4, and enter the following:

- Chart Type – select **Bar**
- Chart Name – enter **Product Reviews**
- Table or View – select **PRODUCT_REVIEWS**
- Label Column – select **PRODUCT_NAME**
- Type – select **Column Value**
- Value Column – select **AVG_RATING**

Add Dashboard Page

Page Name
Dashboard

Set Icon

Chart 1 Chart 2 Chart 3 **Chart 4**

Area Bar Line Pie

Chart Name
Product Reviews

Table or View
PRODUCT_REVIEWS

Label Column
PRODUCT_NAME

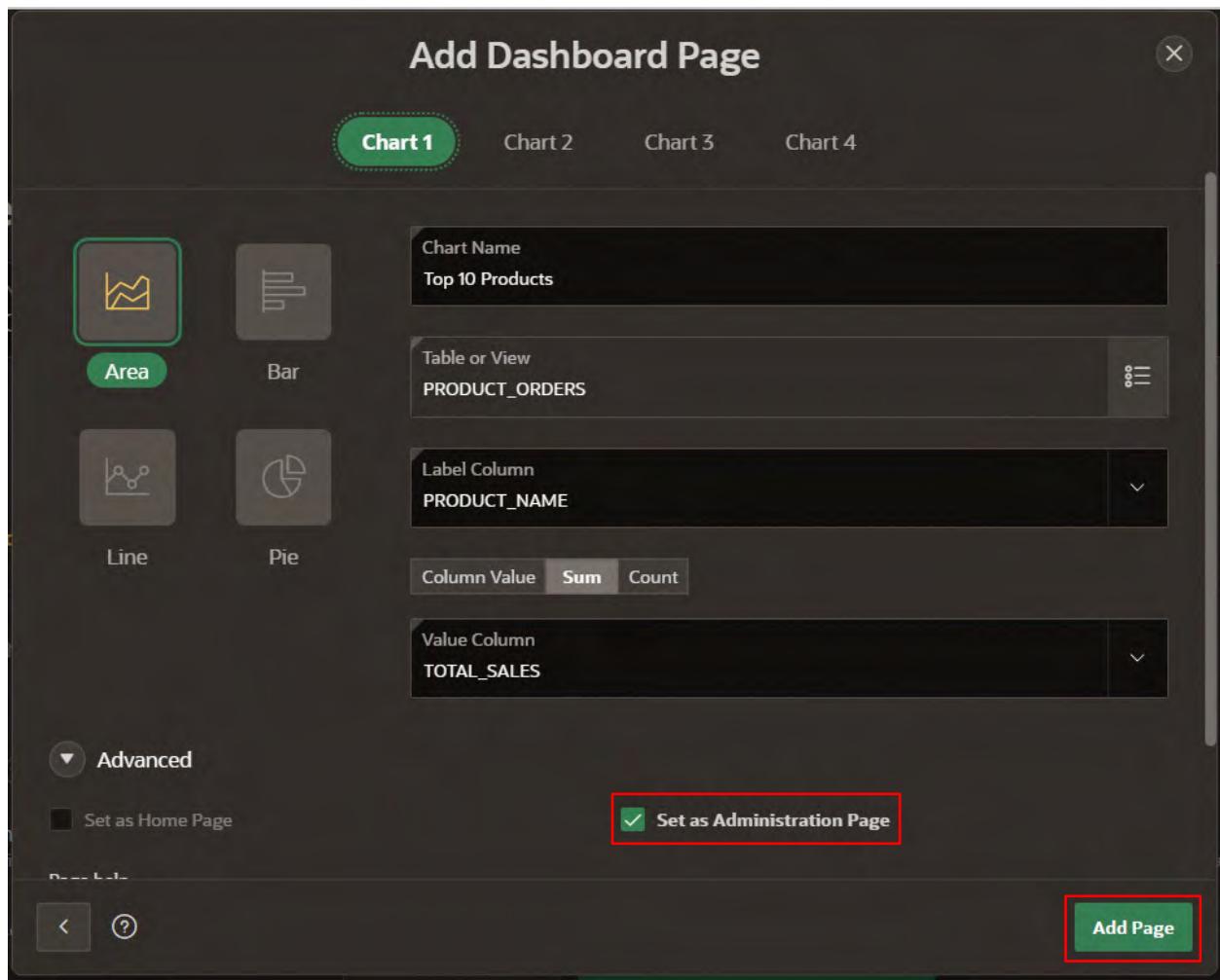
Column Value Sum Count

Value Column
AVG_RATING

Advanced

< ? Add Page

7. Click Advanced and check **Set as Administration Page**.

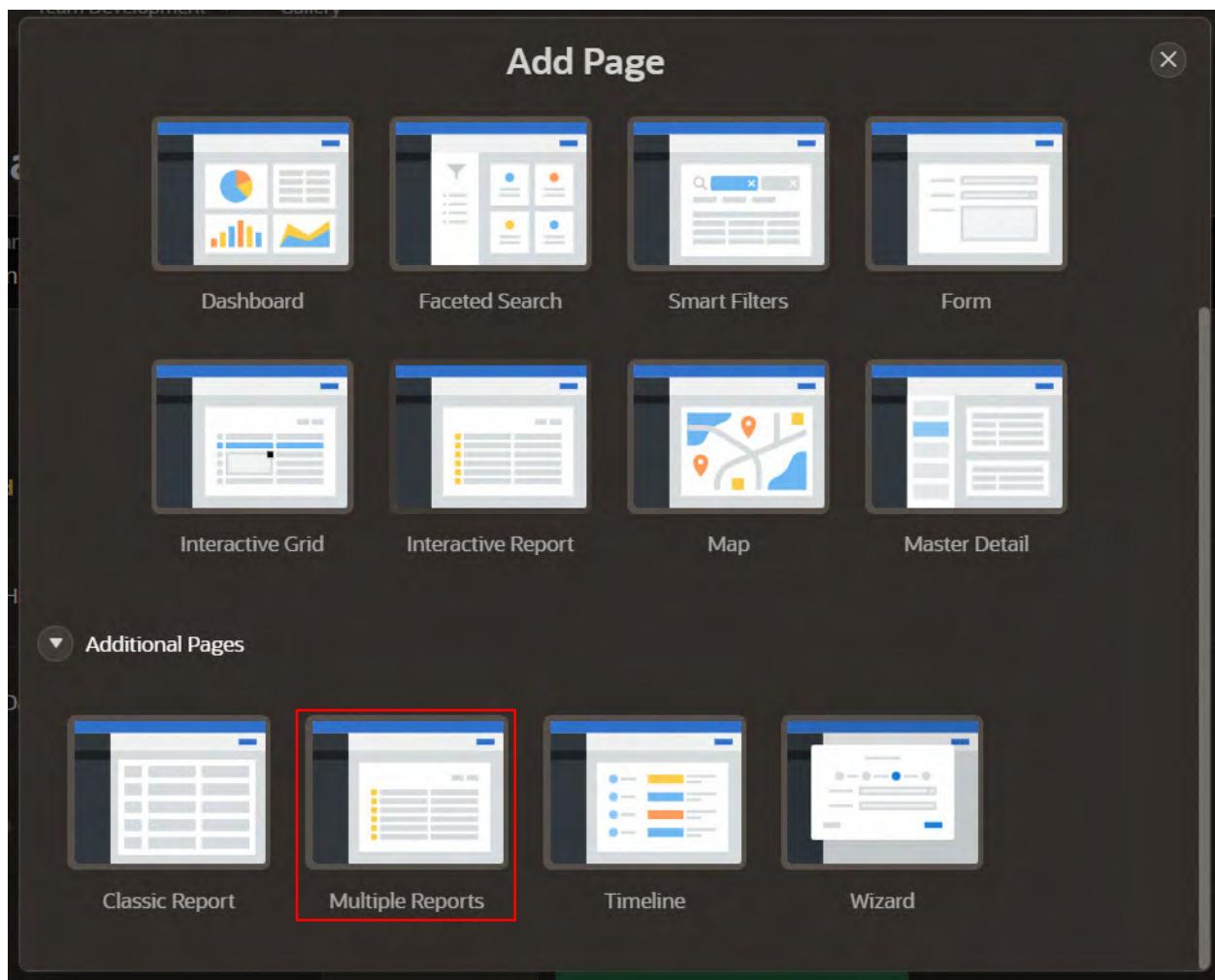


8. Click **Add Page**.

Add Multiple Reports

1. In the Create Application wizard, click **Add Page**.
2. Click the arrow to the left of **Additional Pages** to see additional page types.

3. Click **Multiple Reports**.



4. On the Create Multiple Reports Page, select the following tables:

- CLOTHING_LOOKUP
- COLOR_LOOKUP
- CUSTOMERS
- DEPARTMENT_LOOKUP
- PRODUCT_REVIEWS
- STORES

5. Click **Add Pages**.

Add Multiple Reports

Search: All Text Columns Go X

| <input type="checkbox"/> | Table Name ↑= |
|-------------------------------------|-------------------------|
| <input checked="" type="checkbox"/> | CLOTHING_LOOKUP |
| <input checked="" type="checkbox"/> | COLOR_LOOKUP |
| <input checked="" type="checkbox"/> | CUSTOMERS |
| <input type="checkbox"/> | CUSTOMER_ORDER_PRODUCTS |
| <input type="checkbox"/> | DEPARTMENTS |
| <input checked="" type="checkbox"/> | DEPARTMENT_LOOKUP |
| <input type="checkbox"/> | EBA_DEMO_IR_DEPT |
| <input type="checkbox"/> | EBA_DEMO_IR_EMP |
| <input type="checkbox"/> | EBA_DEMO_IR_PROJECTS |

| | |
|-------------------------------------|---------------------|
| <input type="checkbox"/> | ORDERS |
| <input type="checkbox"/> | ORDER_ITEMS |
| <input type="checkbox"/> | PRODUCTS |
| <input type="checkbox"/> | PRODUCT_ORDERS |
| <input checked="" type="checkbox"/> | PRODUCT_REVIEWS |
| <input type="checkbox"/> | PROJECTS |
| <input checked="" type="checkbox"/> | STORES |
| <input type="checkbox"/> | STORE_ORDERS |
| <input type="checkbox"/> | STORE_ORDERS_STATUS |

6 rows selected Total 20

< ? Add Pages

Set Multiple Reports as Administration Pages

1. Edit each of the following pages to set it as an Administration Page:

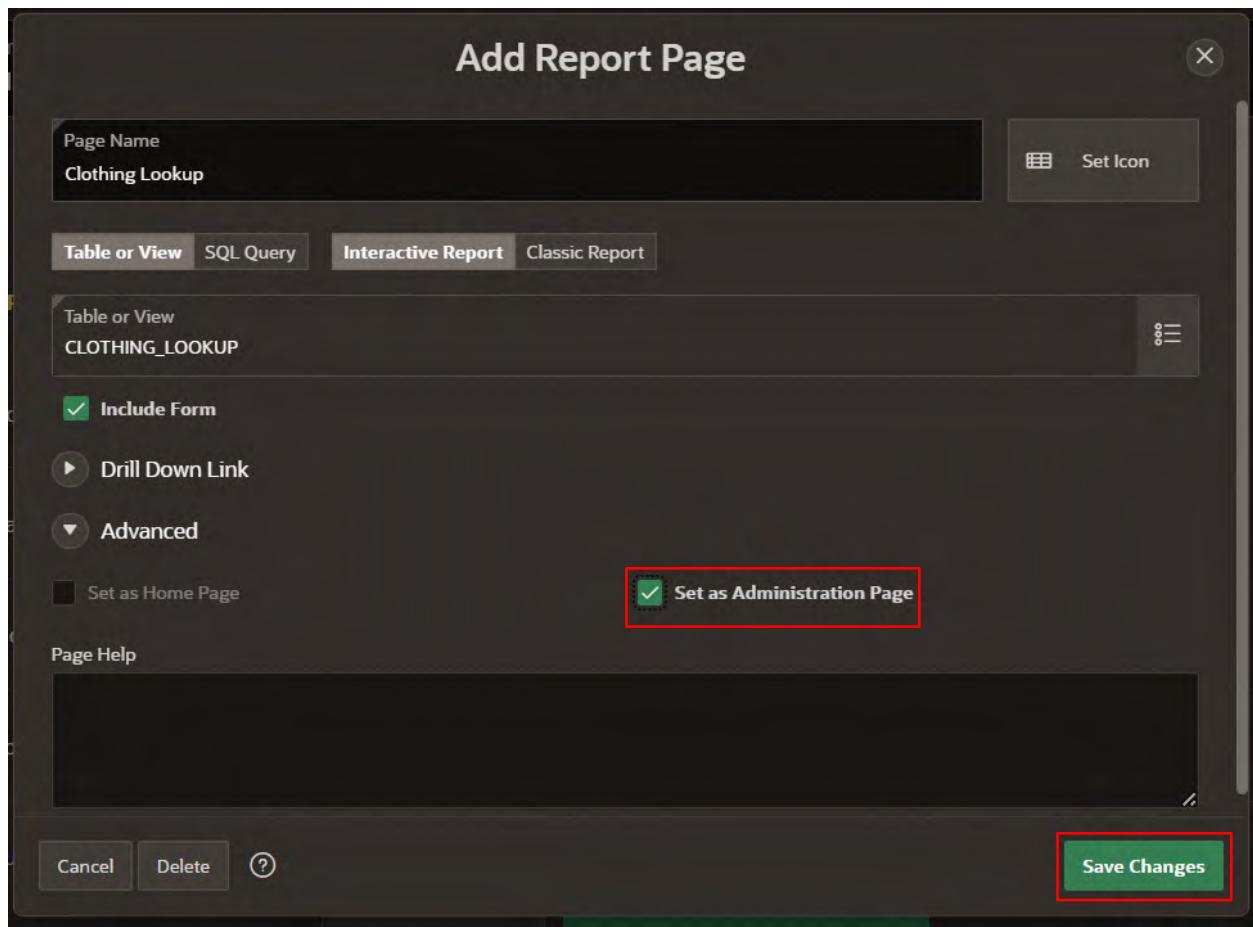
- CLOTHING_LOOKUP
- COLOR_LOOKUP
- CUSTOMERS
- DEPARTMENT_LOOKUP
- PRODUCT_REVIEWS
- STORES

The screenshot shows the 'Create an Application' interface with the following details:

- Name:** Online Shopping Application
- Appearance:** Vita, Side Menu
- Pages:**
 - Home:** Blank
 - Dashboard:** Dashboard (Administration Page)
 - Clothing Lookup:** Interactive Report with Form (clothing_lookup) **Edit** (highlighted with a red box)
 - Color Lookup:** Interactive Report with Form (color_lookup) **Edit**

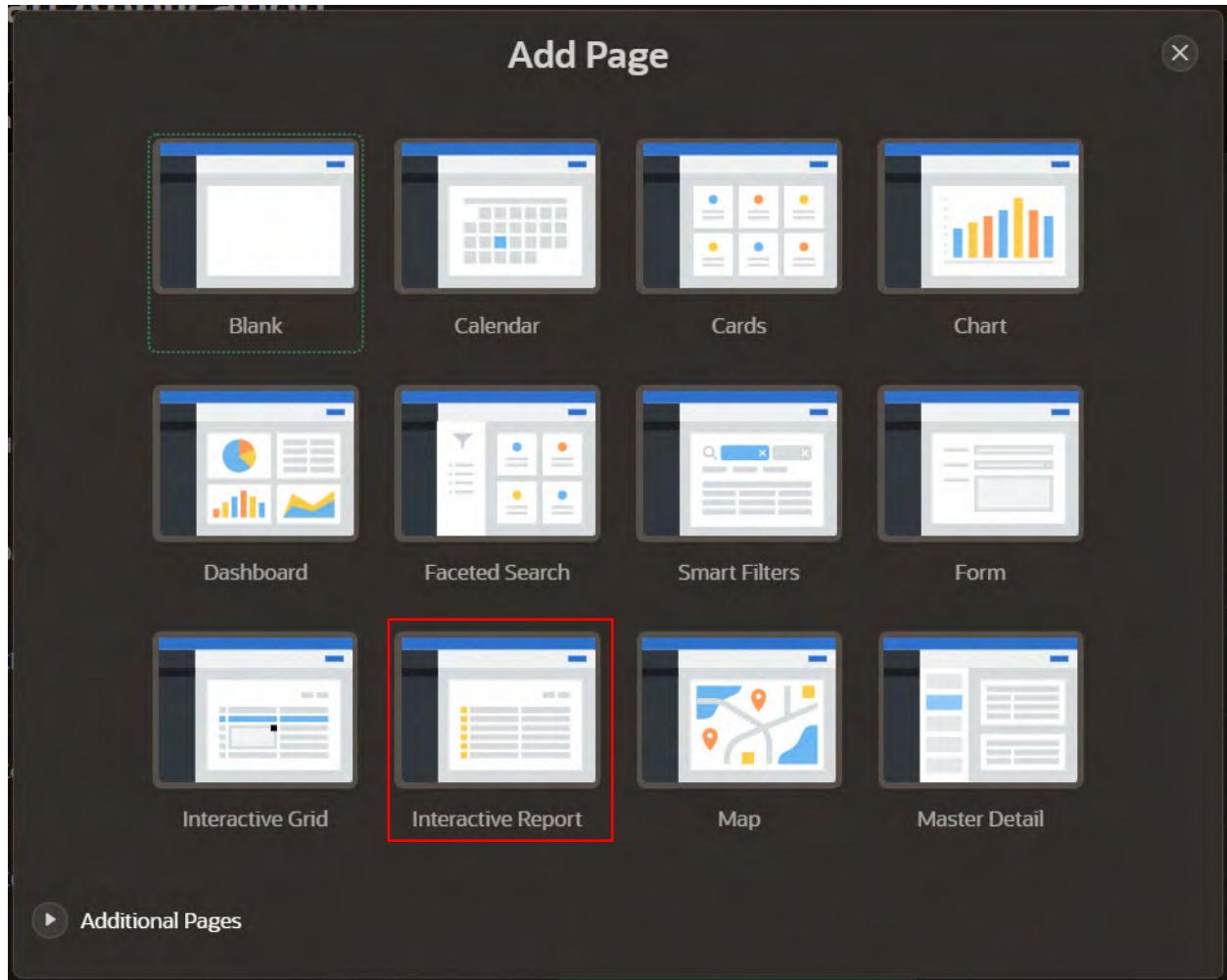
2. For each page you edit, click **Advanced** and check **Set as Administration Page**.

3. Click **Save Changes**.



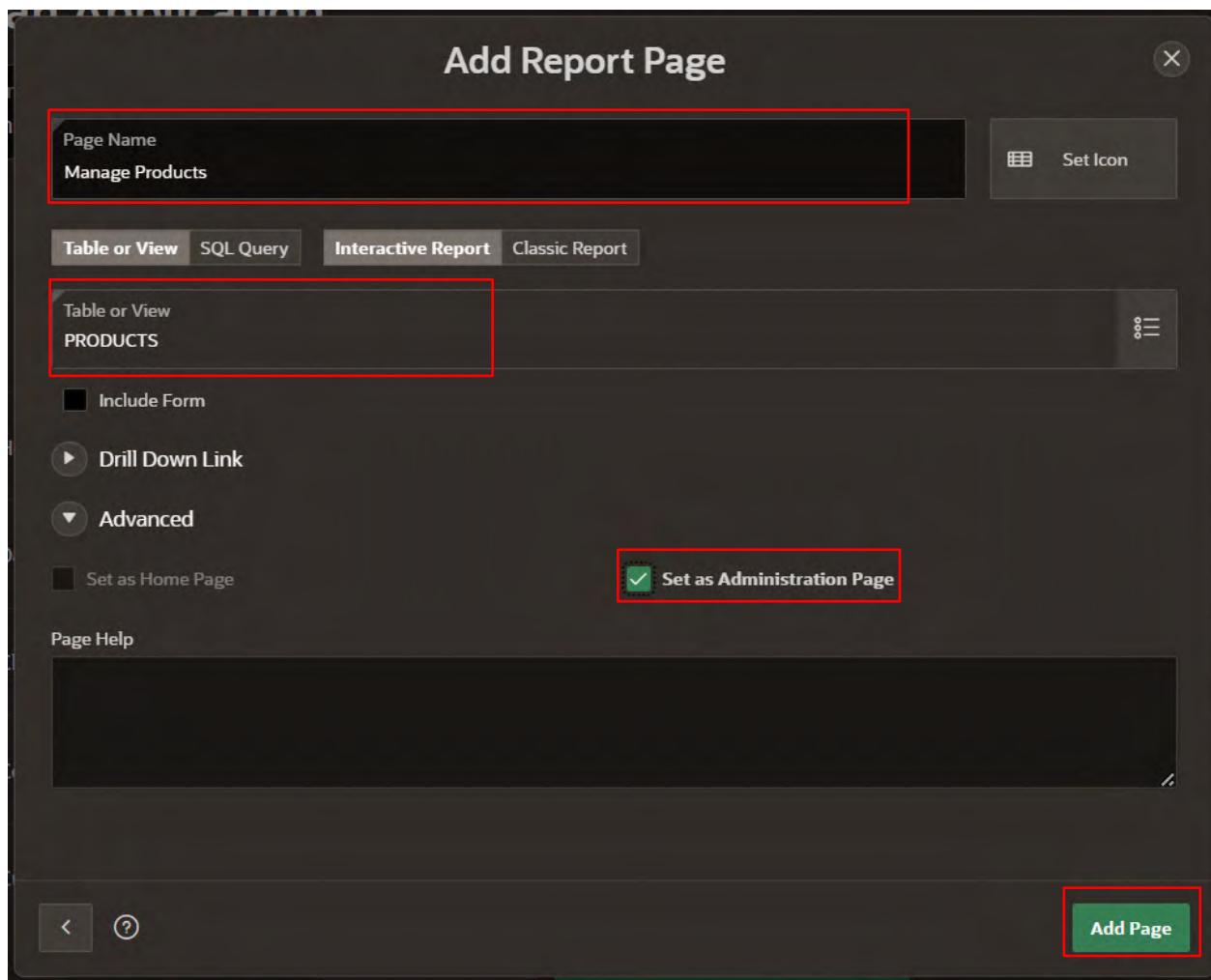
Add Manage Products Page

1. In the Create Application wizard, click **Add Page**.
2. Click **Interactive Report**.



3. On the Report Page, enter the following:
 - Page Name - enter **Manage Products**
 - Table - select **PRODUCTS**
4. Click **Advanced** and check **Set as Administration Page**.

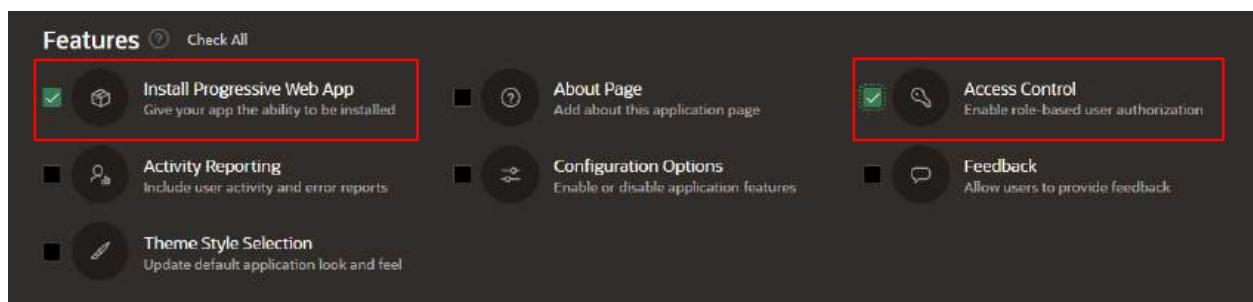
5. Click **Add Page**.



Set Features

Features are a set of optional application capabilities in Oracle APEX that you can include in your new application. Access Control enables role-based user authorization with a single click.

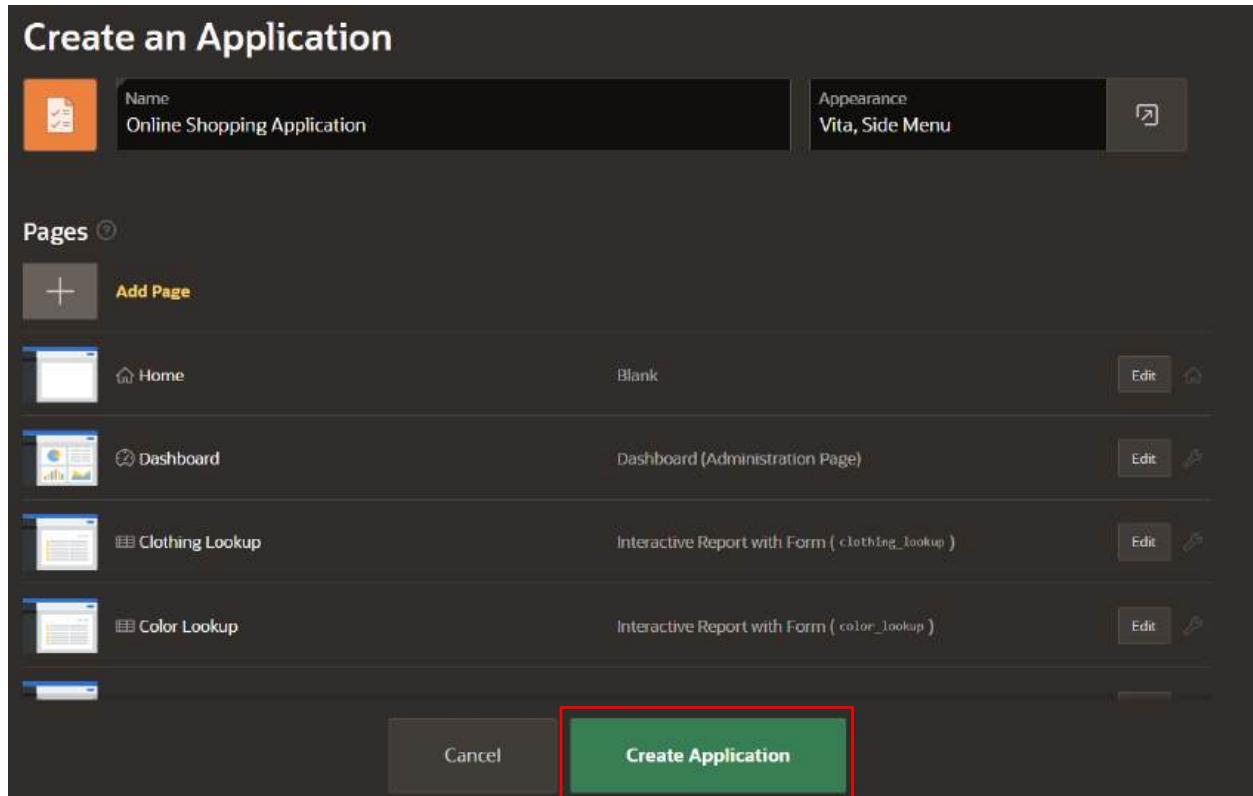
1. Under the **Features** section, check **Install Progressive Web App** and **Access Control**.



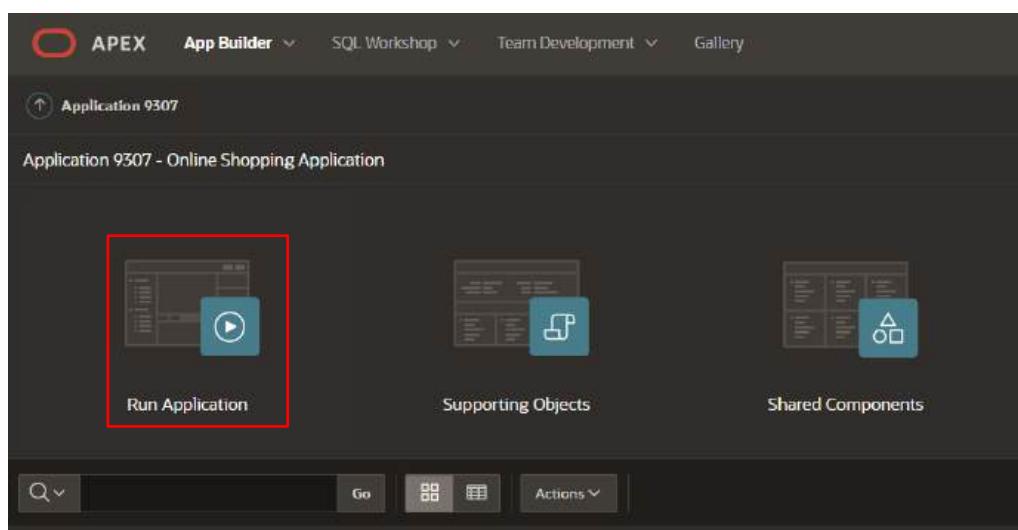
Finish Creating the Application

Now that you have added all the pages, it is time to generate the app and review it.

1. Scroll to the bottom of the page, and click **Create Application**.

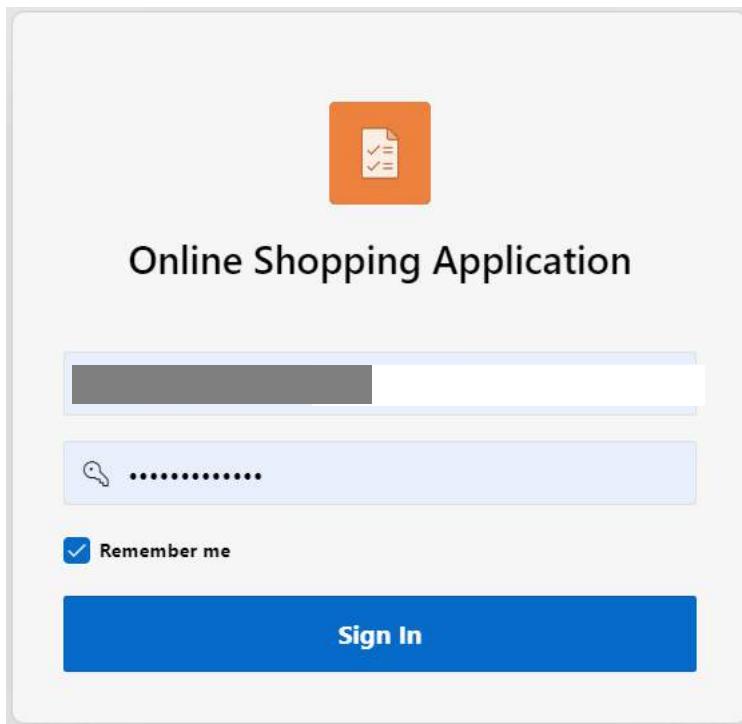


2. Once the application is created, you will find the new app on the application home page. Click **Run Application**.



Run the Application

1. Enter your user credentials. Click **Sign In**.



The new application will be displayed. Explore the pages that you just created by clicking the navigation menu.

You now know how to create an application with a number of different page types based on existing database objects. You may now **proceed to the next lab**.

Practice 2: Create an Application from a Spreadsheet

In this lab, you will learn how to create an APEX application using data imported from a spreadsheet.

Overview

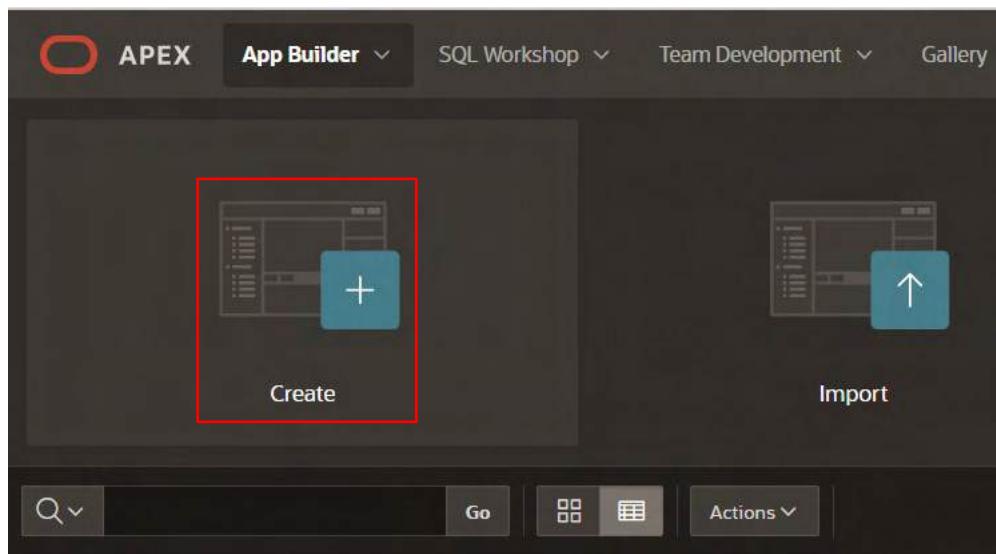
In this lab, you build a simple application based on a spreadsheet. Keep in mind that APEX is great for a variety of apps, from simple ones like this to large, sophisticated apps based on local database objects, REST-enabled SQL objects, and even REST APIs.

While APEX developers spend the majority of their time in the App Builder, you should also investigate the SQL Workshop, where you can create and maintain database objects, Team Development, where you can track large APEX development projects, and the App Gallery, which contains numerous productivity and sample apps that can be installed within minutes.

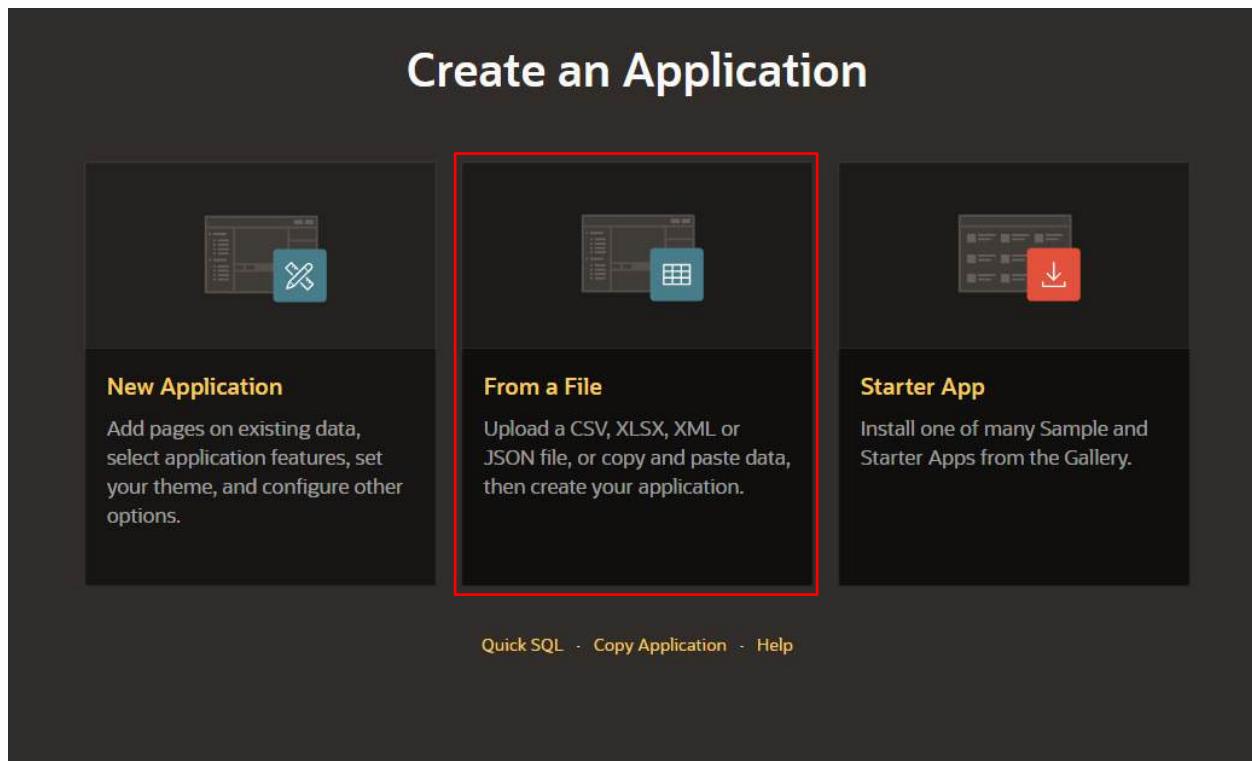
Tasks

Load Projects data

1. From your APEX workspace home page, click **App Builder**.
2. Click **Create a New App**.

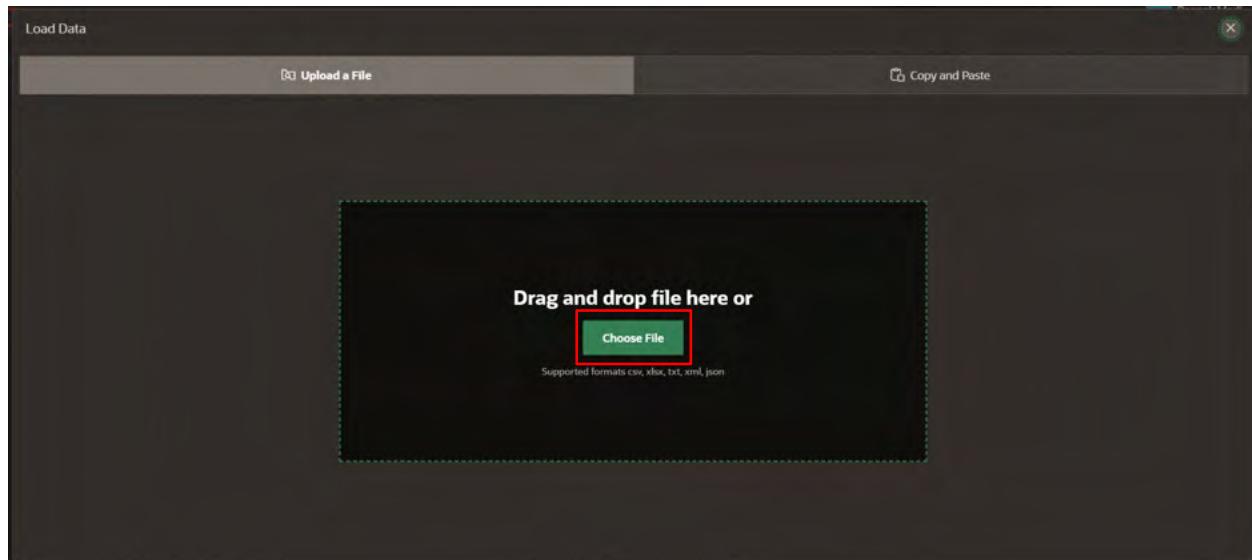


3. Click **From a File**.



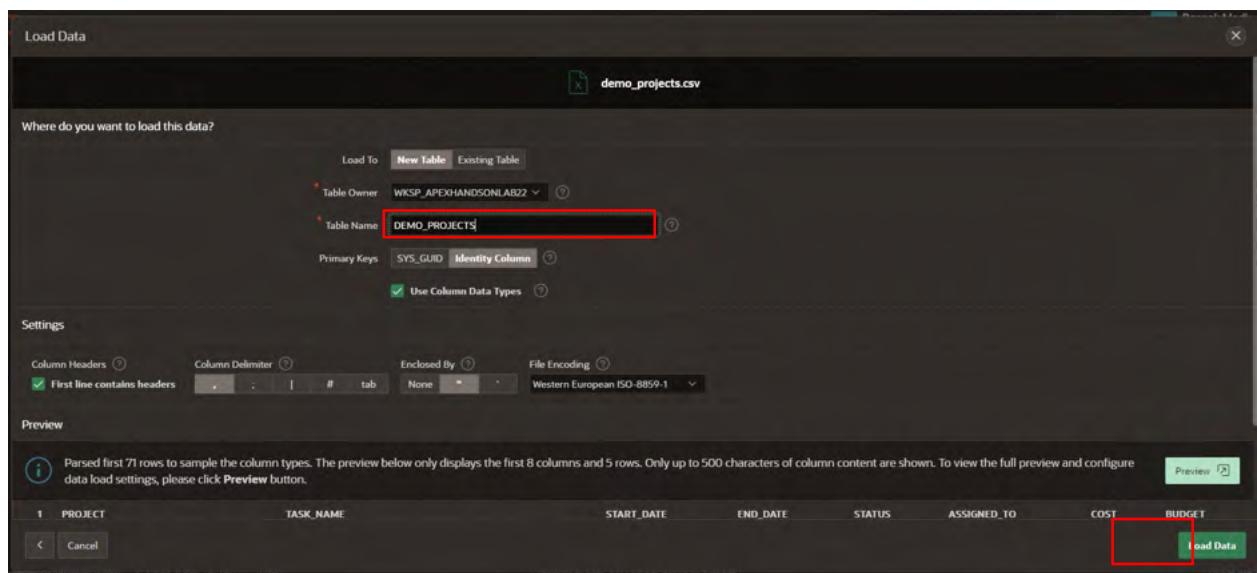
When creating an application from a file, APEX allows you to upload CSV, XLSX, XML, or JSON files and then build apps based on their data. Alternatively, you can also copy and paste CSV data or load sample data.

4. Within the Load Data wizard, click the **Choose File** button.



5. Download and select the [demo_projects.csv](#) file from your local System

6. Review the parsed data. Enter **DEMO_PROJECTS** for Table Name and click **Load Data**.

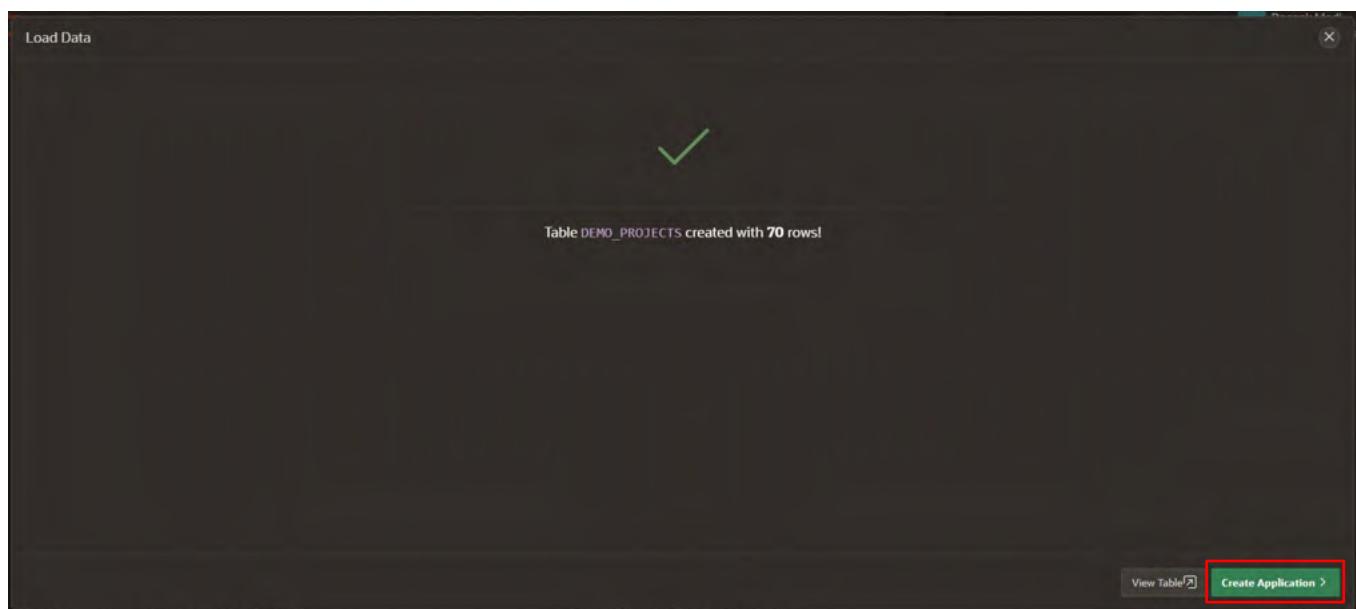


After clicking **Load Data**, you will see a spinner until the wizard finishes loading the data. Continue to Task 2 at this point.

Create an application

The Data Load wizard has created a new table and populated that table with the records from the sample data. Now you can create an app based on this new table.

1. In the Load Data dialog, verify that 70 rows have been loaded into the **DEMO_PROJECTS** Table, then click **Create Application**.



2. Now you are going to remove the following pages from the Create an Application page.
- Demo Projects Search
 - Demo Projects Report
 - Calendar On the Create Application page. Click the **Edit** button next to **Demo Projects Search**.

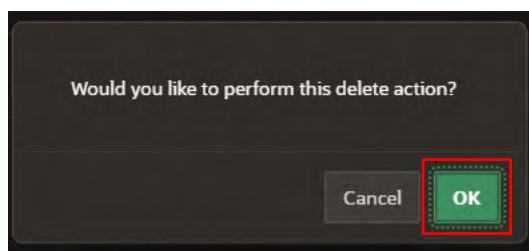
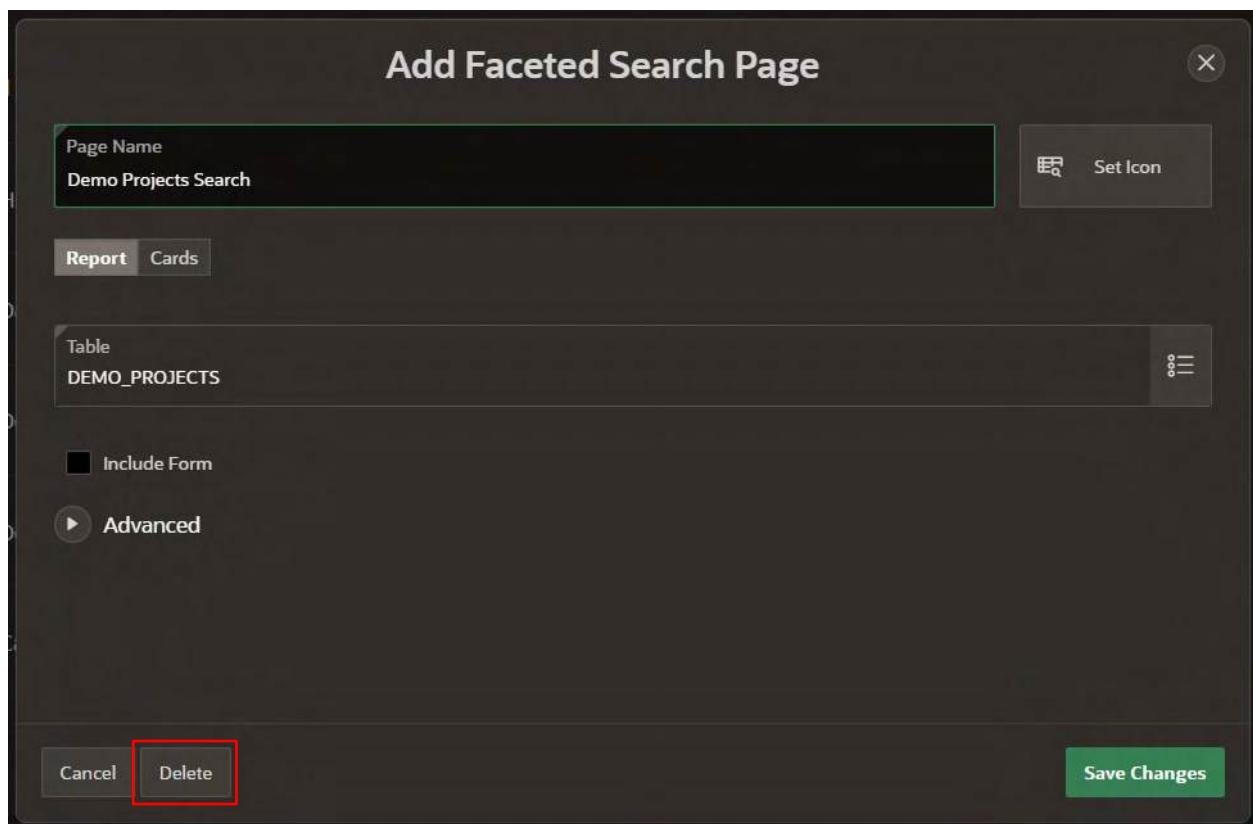
Create an Application

Name: Demo Projects | Appearance: Vita, Side Menu

Pages ?

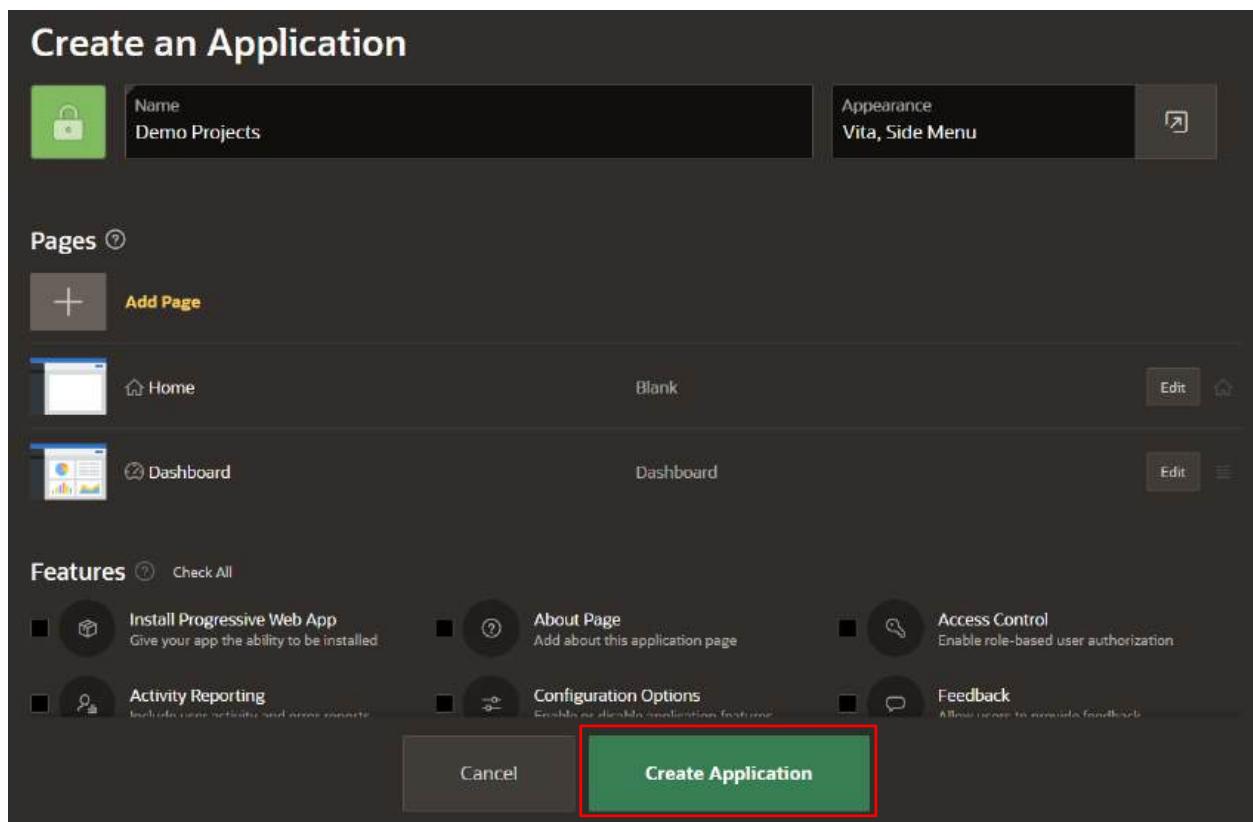
| Add Page | Page Name | Type | Action |
|----------|----------------------|--|--|
| + | Home | Blank | Edit |
| | Dashboard | Dashboard | Edit |
| | Demo Projects Search | Faceted Search (demo_projects) | Edit |
| | Demo Projects Report | Interactive Report with Form (demo_projects) | Edit |

3. On the Add Faceted Search Page, click the **Delete** button and then confirm it by clicking **OK**.



Repeat the steps for **Demo Projects Report** and **Calendar**.

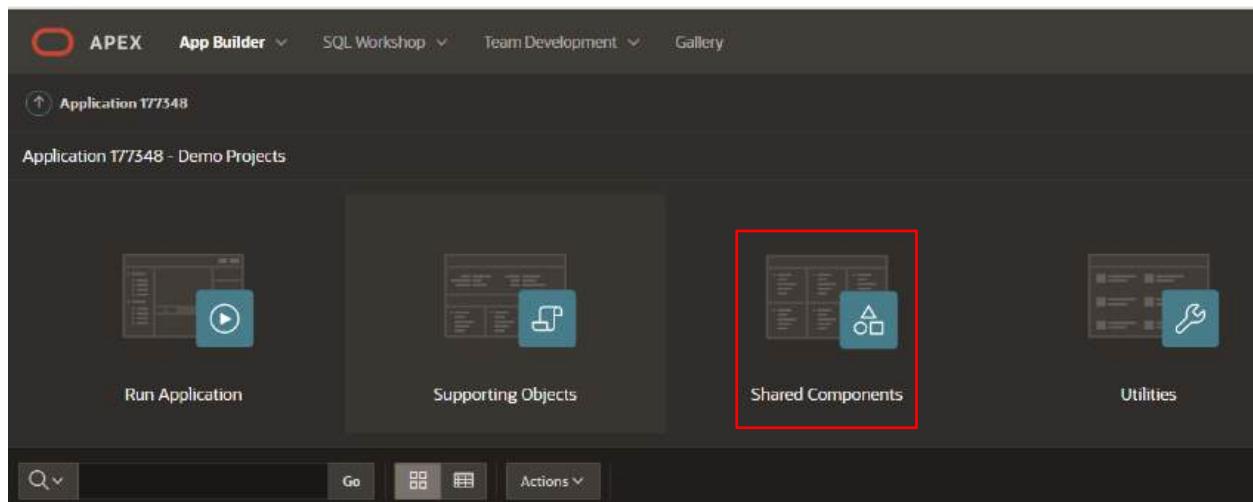
4. Then click **Create Application**.



When the wizard finishes creating the application, you will be redirected to the application's home page in the App Builder.

Configure Progressive Web App Attributes

1. Navigate to the Edit Application Definition page. From the Application home page, you can access the Edit Application Definition page using **Shared Components**.



2. On the Shared Components page, under User Interface, select **Progressive Web App**.

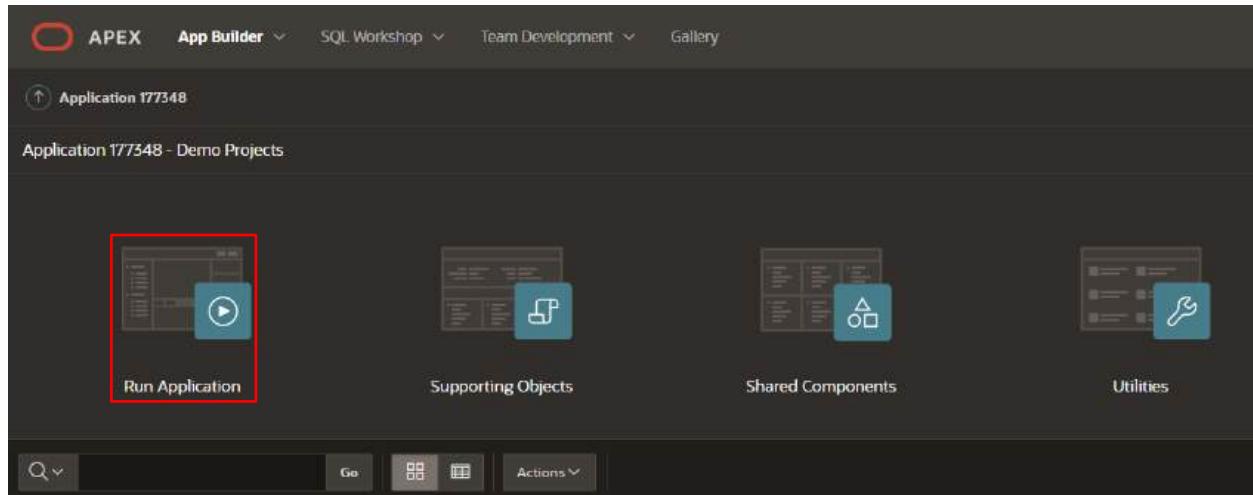


3. Now, navigate to **Progressive Web App** and set **Installable** to **Yes**. Click **Apply Changes**.

The screenshot shows the 'Progressive Web App' configuration page for Application 177348. At the top, there's a navigation bar with tabs for 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this, the breadcrumb path shows 'Application 177348 \ Progressive Web App'. The main content area has tabs for 'Definition', 'Security', 'Globalization', 'User Interface', and 'Progressive Web App', with 'Progressive Web App' highlighted by a red box. The 'General' tab is selected. It contains settings for enabling the PWA and setting it as installable. The 'Installable' switch is highlighted with a red box. Other settings include 'Display' (Standalone), 'Screen Orientation' (Any), 'Theme Color' (Default), 'Background Color' (Default), and 'iOS Status Bar Style' (Translucent). The background is dark grey.

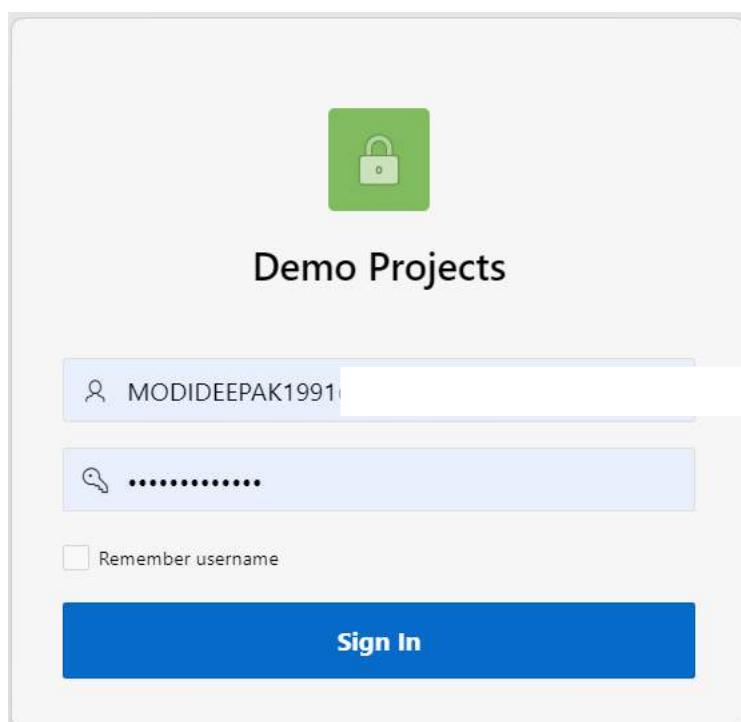
Run and Explore the New App

1. Navigate back to **DEMO_PROJECTS** application and then click **Run Application**. This will open the runtime application in a new browser tab, allowing you to see how end users will view the app.



2. Enter your user credentials and click **Sign In**.

Note: Use the same Username and Password you used to sign in to the APEX Workspace.



3. Click the **Install App** icon on the top right of the page to install the application to your local system.



You now know how to create an application from a spreadsheet by either dragging and dropping a file or loading sample data for training purposes.

Practice: Working with Pages and Regions

Practice 1: Create Pages, Regions, and Items

Overview

In this lab, you open the home page in page designer, and navigate through and review the page designer panes. Then, you create a new page that allows customers to view the details of their recent order. Customers will find the following details of the order:

- Order number
- Order date
- Status
- Total price
- Quantity and price of the items.

In this lab, you will:

1. Navigate and review the Page Designer panes
2. Create a page to review the items that customer just bought

Downloads:

- Did you miss out trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

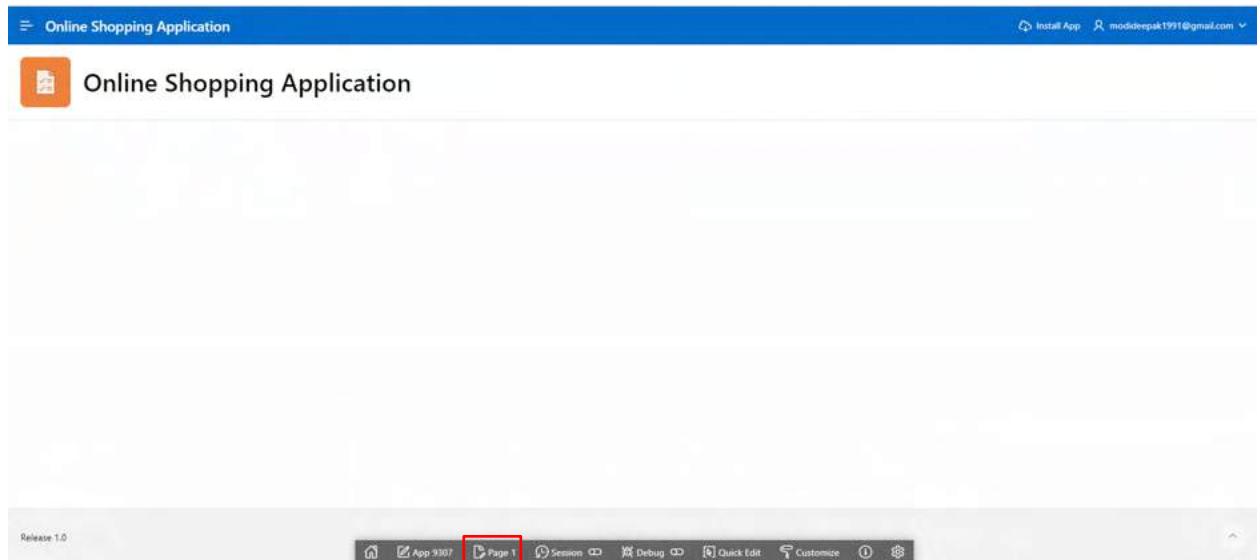
Tasks

Navigate and Review the Page Designer panes

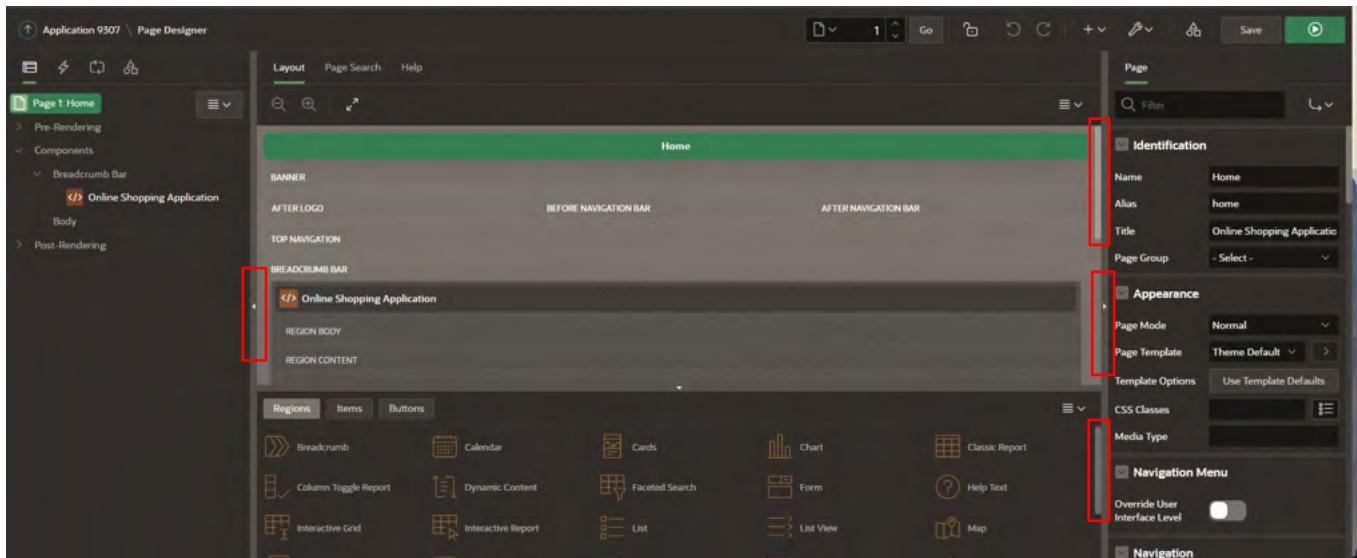
Open the home page in page designer, and navigate through and review the page designer panes.

1. In the App Builder, run the Online Shopping Application. You are now in the application runtime environment. Use the application menu to navigate to the Home page. In the Developer Toolbar, click **Page n**.

Note: If you are not on the Home page then the Developer Toolbar will show the current page number, and clicking Edit Page xx will navigate to that page, instead of Page n.

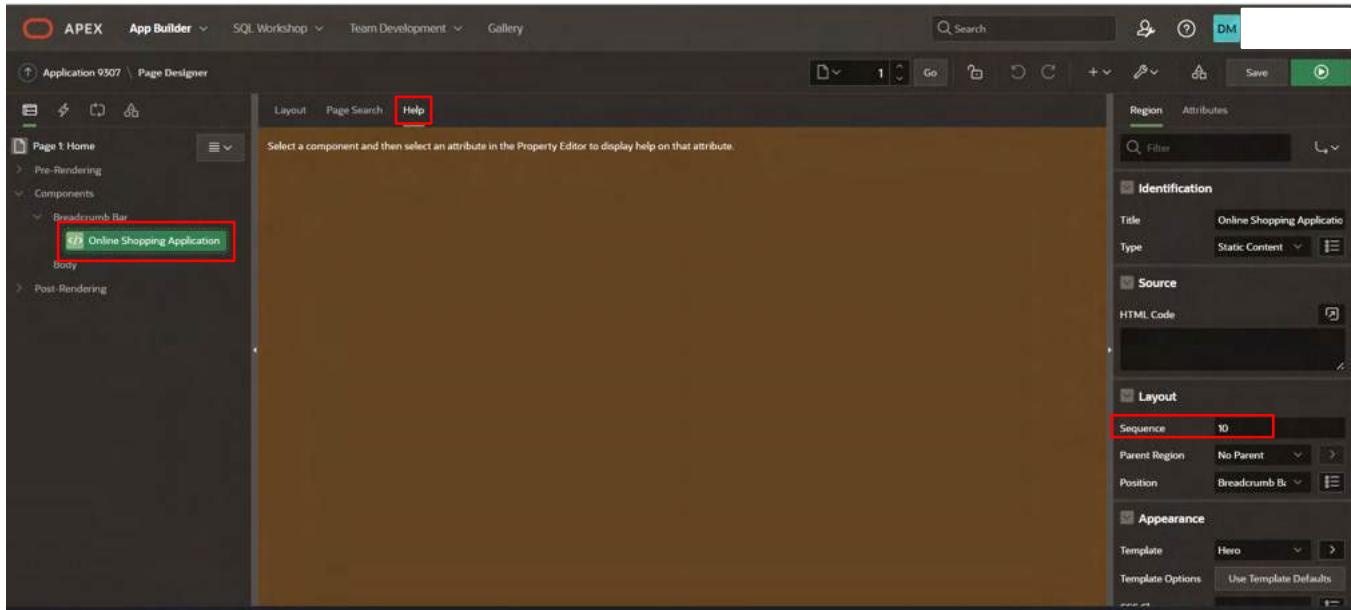


2. The Page Designer is displayed for Page 1. There are three main panes within Page Designer: Left Pane, Central Pane, and Right Pane. You can change the size of each pane by selecting the dividers and sliding them left or right. Change the size of Grid Layout and Gallery by sliding the divider between them up and down.

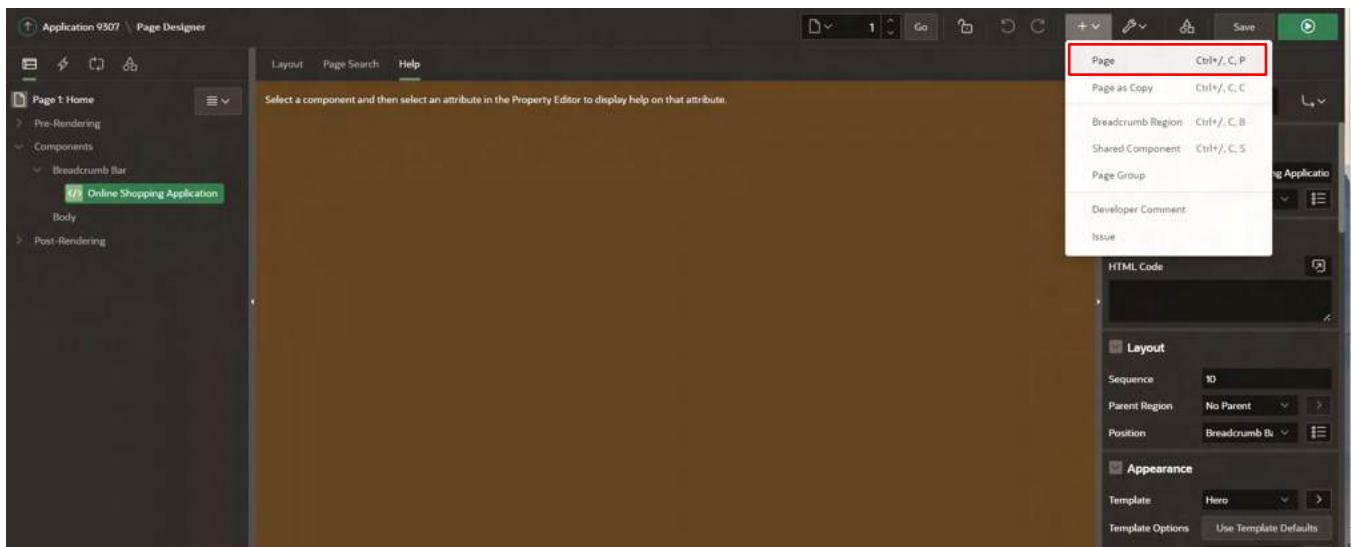


3. In the Page Designer, you can invoke help on any attribute by clicking **Help** on the toolbar. Select a component and then select an attribute in the Property Editor to display help on that attribute.

For example, in this screenshot, you select **Online Shopping Application** in the Rendering tree on the left, and then select **Sequence > Layout** in the Property Editor on the right. The help text for **Sequence** is displayed.



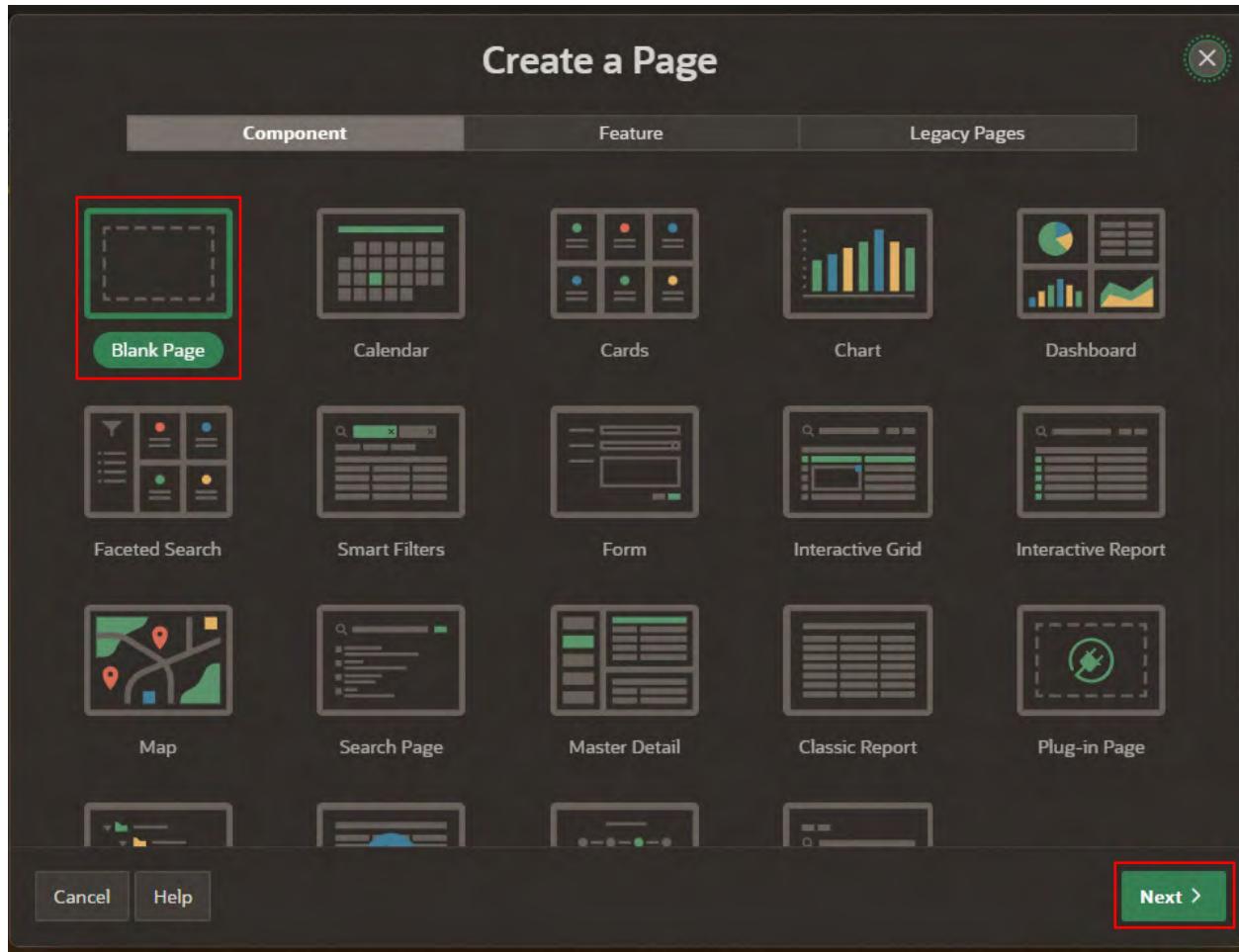
4. In the page designer, click Create (the + icon) and select **Page**.



Create a Normal Page - Order Information

Create a Normal Page to review the order that the customer has made.

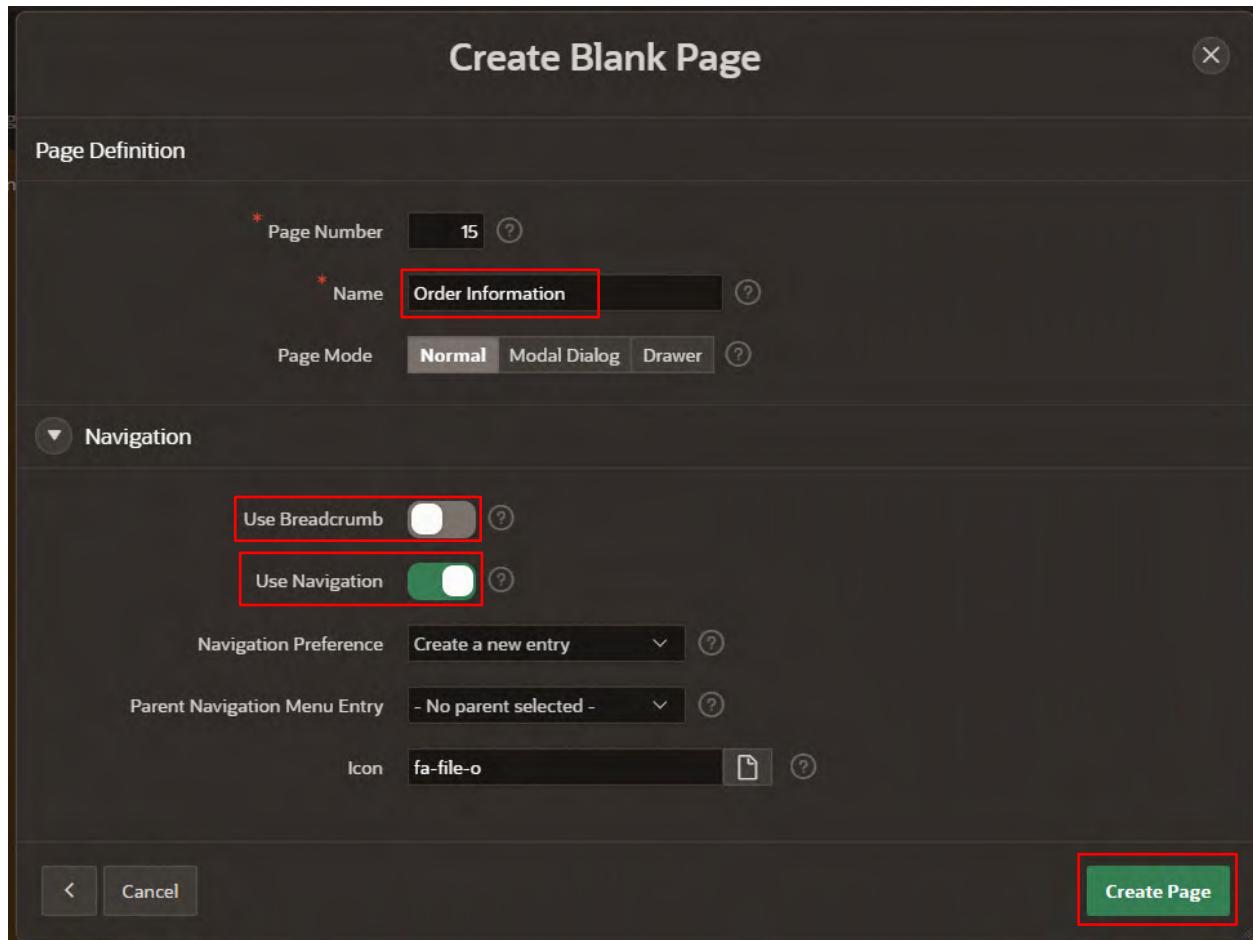
1. In the Create a Page dialog, select **Blank Page** and click **Next**.



2. Enter the following:
 - Page Number - enter **15**
 - Name - enter **Order Information**

Now, click **Navigation**, and enter the following.

- Breadcrumb - Set to **No**
- Navigation - Set to **Yes**. Click **Create Page**.

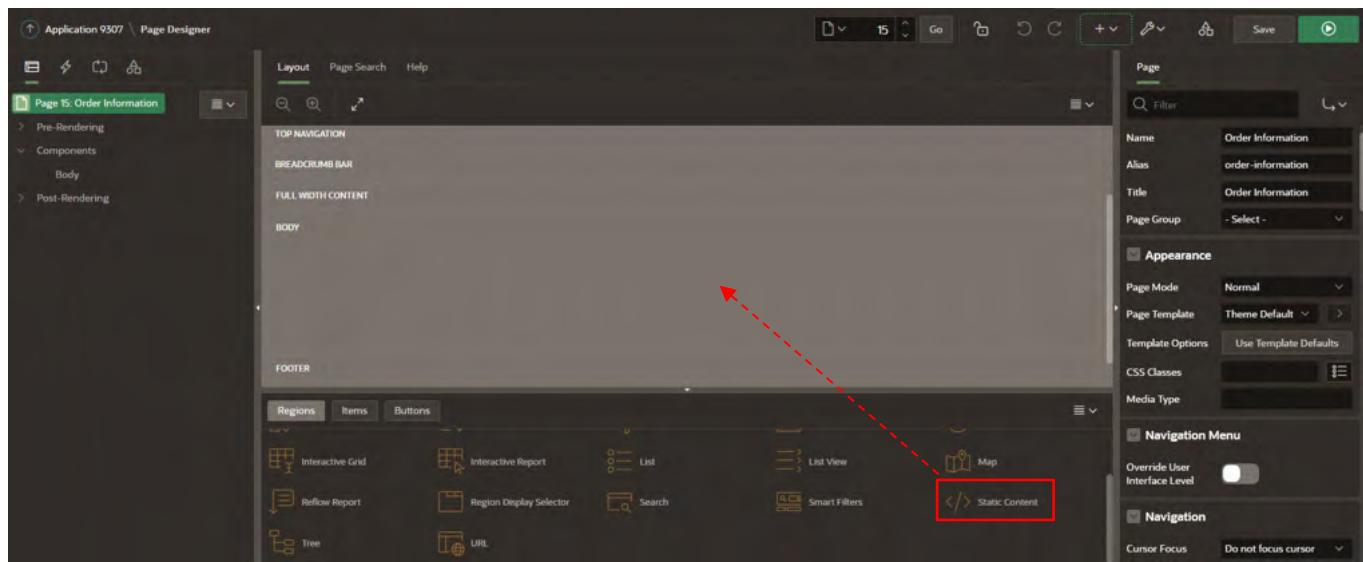


Add a Region

Add a region to the page to display order details.

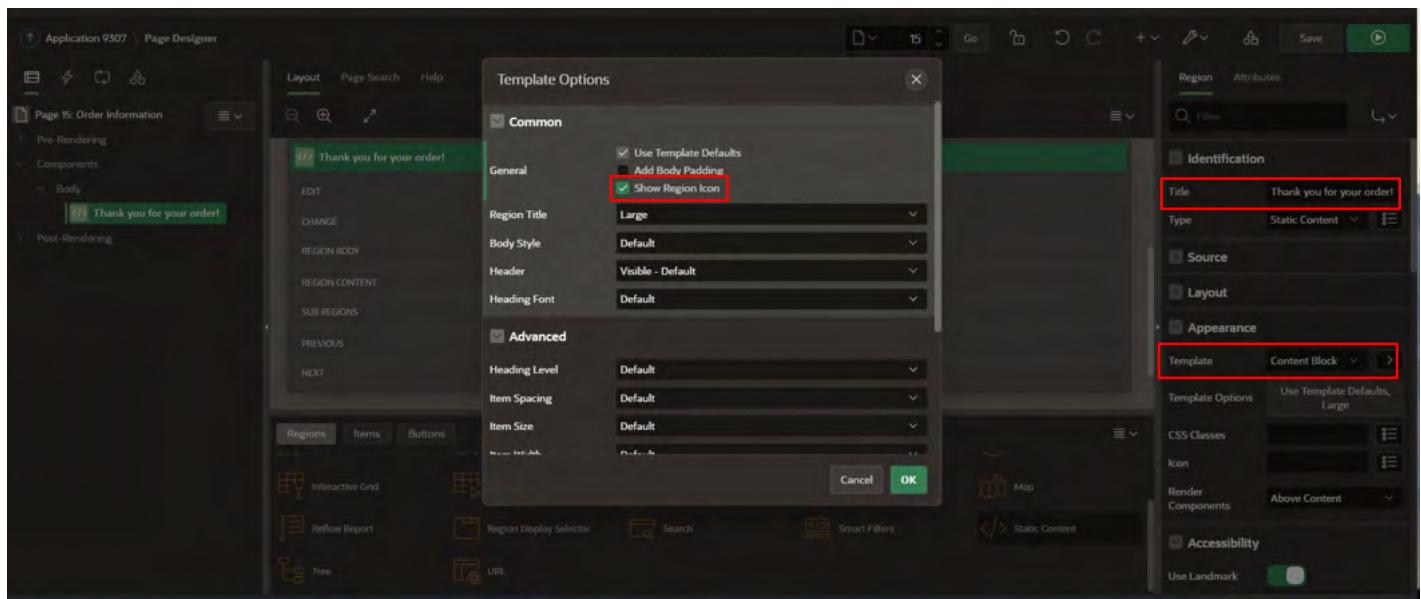
1. On the new page created, navigate to the **Gallery Menu** at the bottom of the page showing Regions, Items, and Buttons categories and ensure that **Regions** is selected.

2. Drag a **Static Content** region and drop it to the **Body** section.

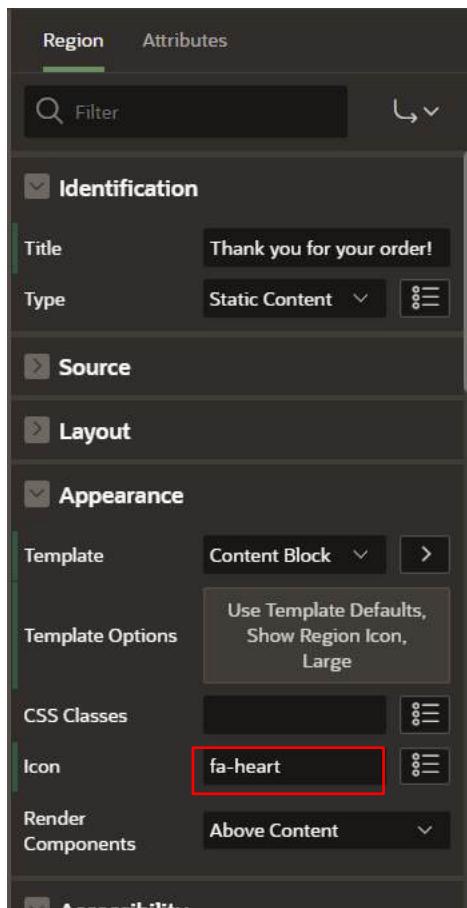


3. In the Property Editor, enter the following:

- For Title - enter Thank you for your order!
- For Template - select **Content Block**
- For Template Options - Click the rectangular area next to the Template Options label to open the Template Options dialog. On the Dialog page, select **Show Region Icon** and click **OK**.



- For Icon, enter **fa-heart**.

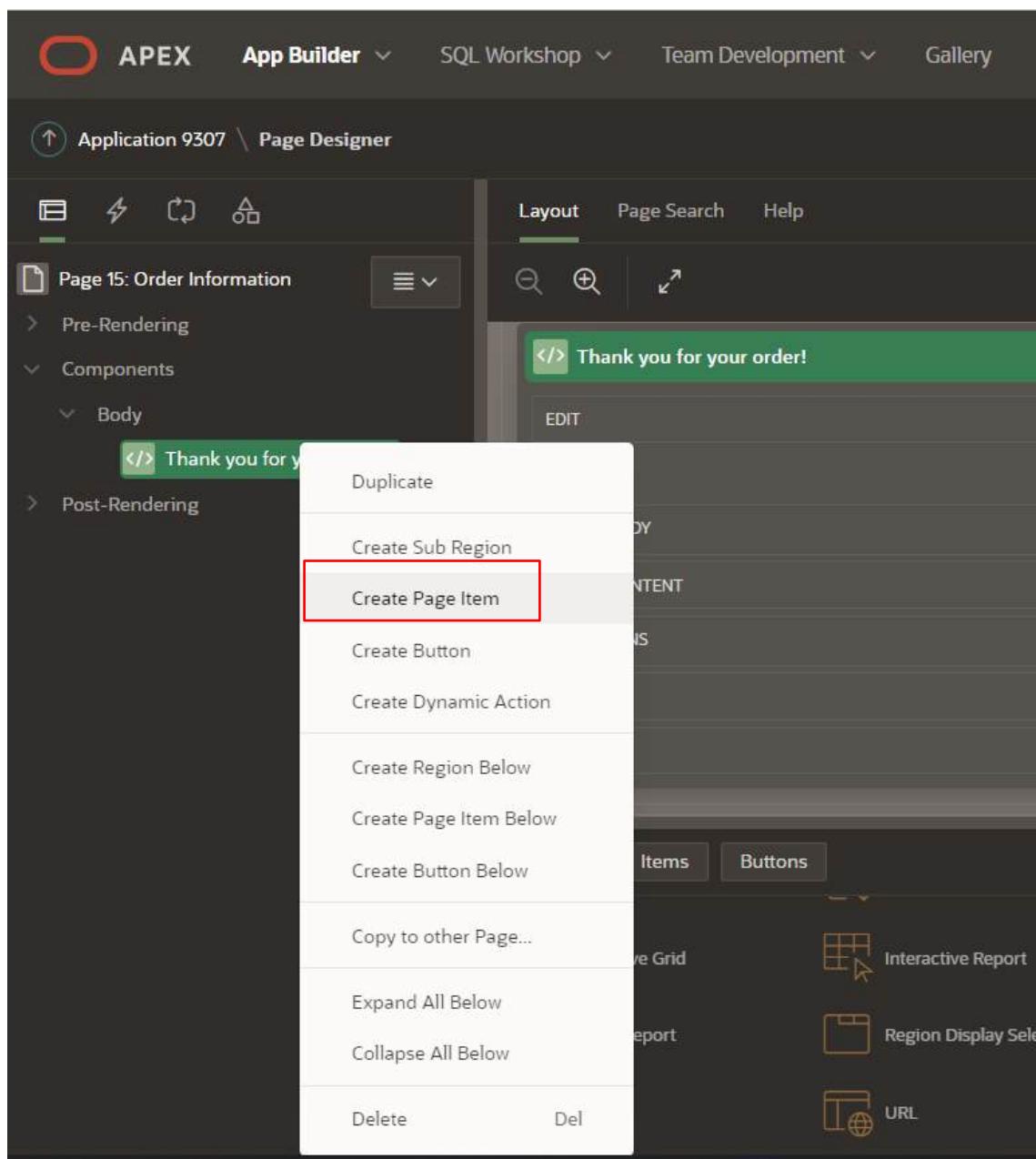


Add Items to the Page

Add a hidden item to store the order ID without the user being able to see it.

1. In the Rendering tree (left pane), click the **Thank you for your order!** region to select it.

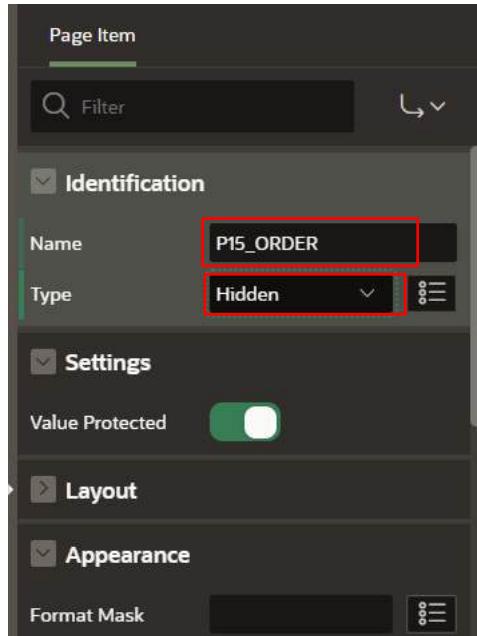
2. Right-click the region and click **Create Page Item**.



- In the property editor, set the name and type as follows:

Table 1: Work with Pages and Regions | Lab 1: Create Pages, Regions and Items

| Name | Type |
|-----------|--------|
| P15_ORDER | Hidden |

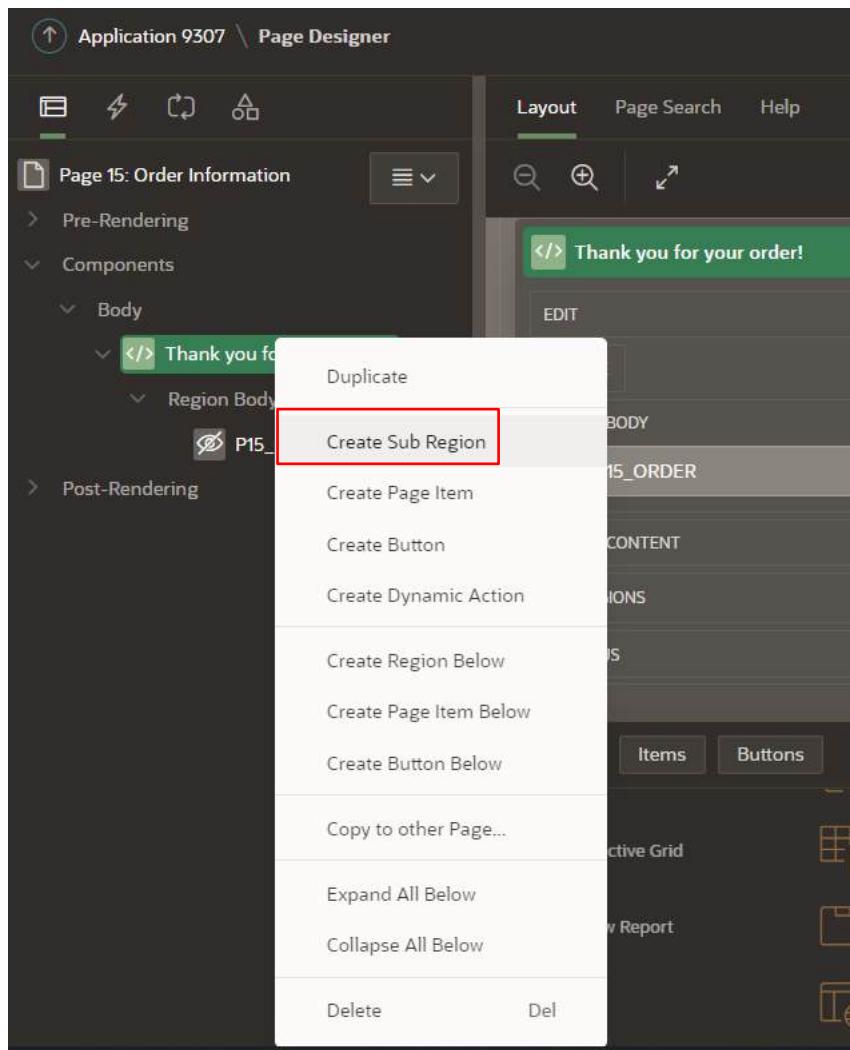


Add Static Content Region

Add a region to contain Order details and items.

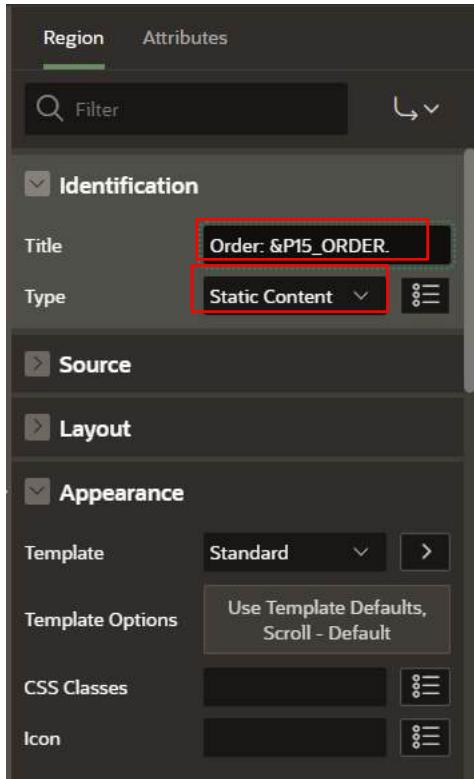
- In the Rendering tree (left pane), click the **Thank you for your order!** region.

2. Right-click the region and click **Create Sub Region**.



3. In the Property Editor, enter the following:

- For Title - enter the expression (including the period) **Order: &P15_ORDER.**
- For Type - select **Static Content**

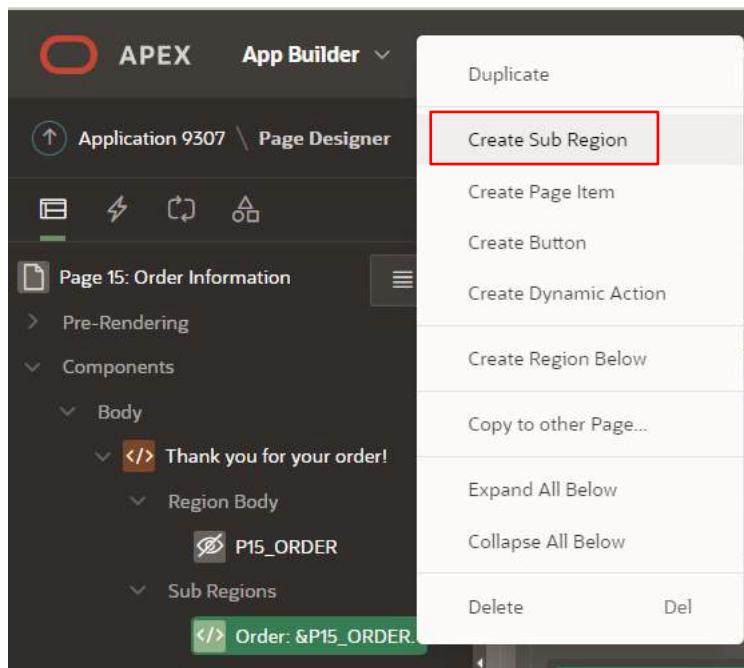


Add Order Details Region

Add a region to display Order details.

1. In the Rendering tree (left pane), navigate to the **Order: &P15_ORDER.** region.

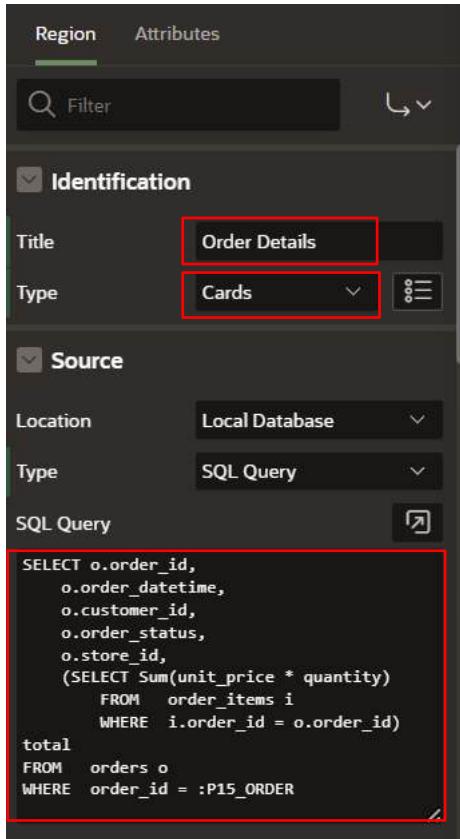
2. Right-click the region and click **Create Sub Region**.



3. In the Property Editor, enter the following:

- For Title - enter **Order Details**
- For Type - select **Cards**
- Under Source section:
 - For Type - select **SQL Query**
 - For SQL Query - enter the following SQL Query:

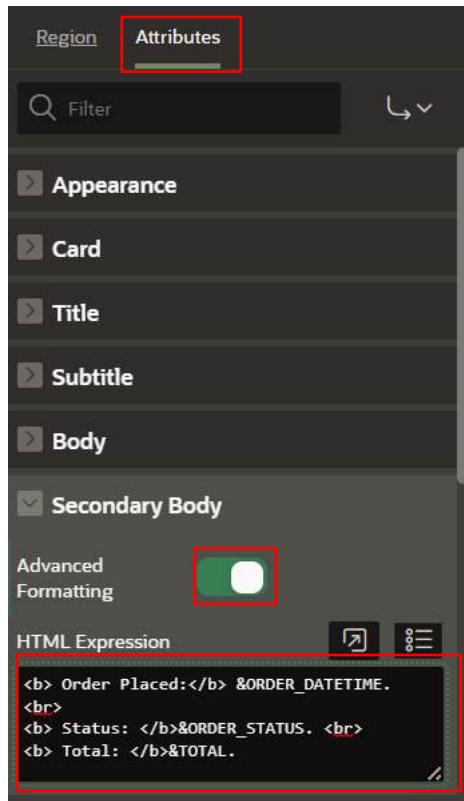
```
SELECT o.order_id,
       o.order_datetime,
       o.customer_id,
       o.order_status,
       o.store_id,
       (SELECT Sum(unit_price * quantity)
        FROM   order_items i
        WHERE  i.order_id = o.order_id) total
  FROM   orders o
 WHERE  order_id = :P15_ORDER
```



4. Click **Attributes**.

- Search for Secondary Body in the filter and do the following:
 - Set Advanced Formatting to **On**
 - For HTML Expression - enter:

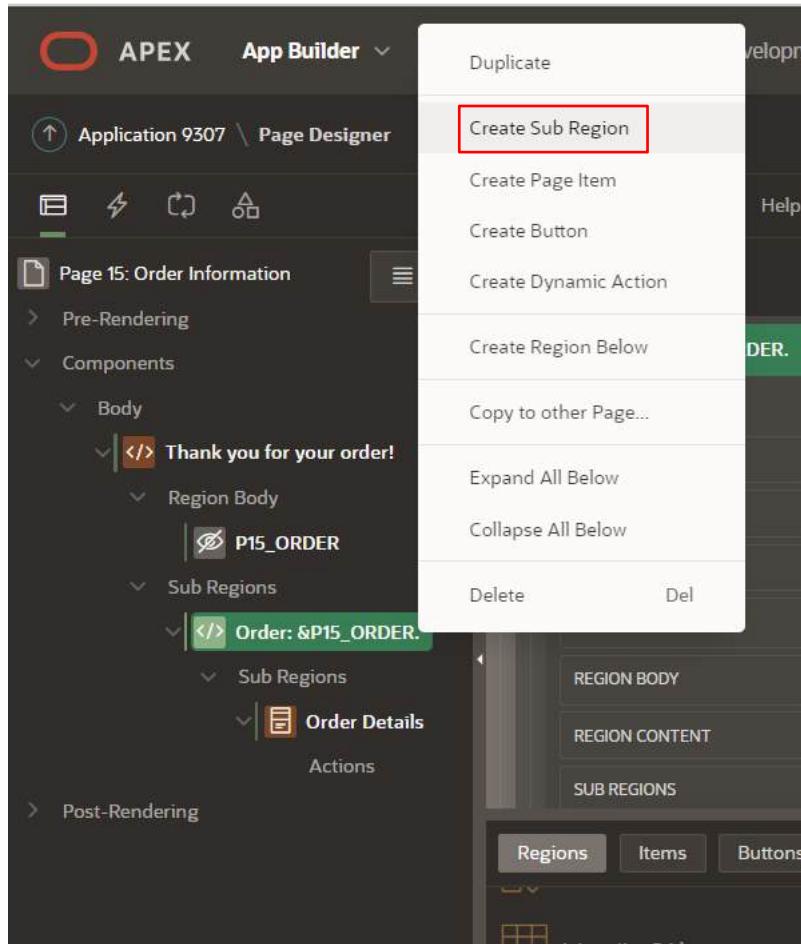
```
<b> Order Placed:</b> &ORDER_DATETIME. <br>
<b> Status:</b> &ORDER_STATUS. <br>
<b> Total:</b> &TOTAL.
```



Add Items Region

Add a region to display items in the order.

1. In the Rendering tree (left pane), navigate to the **Order: &P15_ORDER**. region.
2. Right-click the region and click **Create Sub Region**.

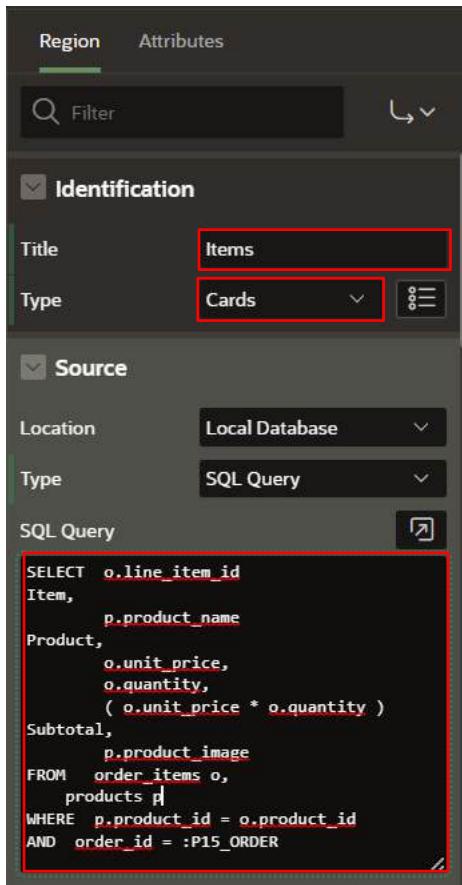


3. In the Property Editor, enter the following:

- For Title - enter **Items**
- For Type - select **Cards**
- Under Source section:
 - For Type - select **SQL Query**
 - For SQL Query - enter the following SQL Query:

```

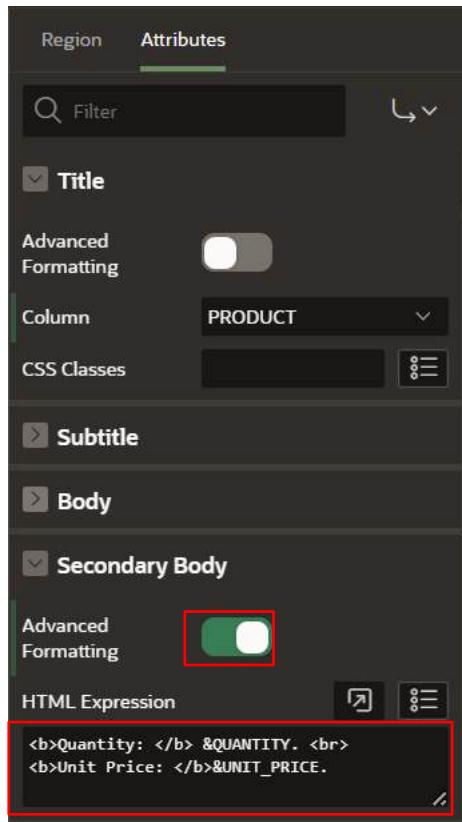
SELECT    o.line_item_id           Item,
          p.product_name        Product,
          o.unit_price,
          o.quantity,
          ( o.unit_price * o.quantity ) Subtotal,
          p.product_image
FROM      order_items o,
          products p
WHERE    p.product_id = o.product_id
AND      order_id = :P15_ORDER
  
```



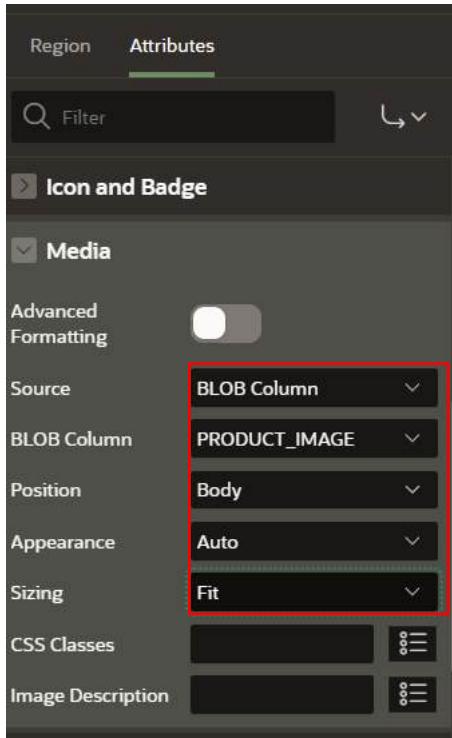
4. Click **Attributes** and do the following:

- Under Title section:
 - For Column - select **PRODUCT**
- Under Secondary Body:
 - Set Advanced Formatting to **On**
 - For HTML Expression - enter:

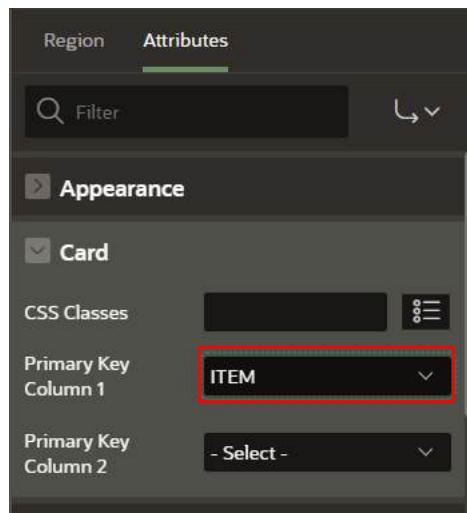
```
<b>Quantity: </b> &QUANTITY. <br>
<b>Unit Price: </b>&UNIT_PRICE.
```



- Under Media section:
 - For Source - select **BLOB Column**
 - For BLOB Column - select **PRODUCT_IMAGE**
 - For Position - select **Body**
 - For Appearance - select **Auto**
 - For Sizing - select **Fit**



- Under Card:
 - For Primary Key Column - Select **ITEM**.



5. Click **Save**.

You now know how to add a new page to your existing APEX Application and add regions to define the page's content using the Page Designer. You may now **proceed to the next lab**.

Practice: Developing Reports

Practice 1: Create an Interactive Grid and Smart Filters

Overview

In this practice, you navigate to the **Demo Projects** application and create an Interactive Grid page and a Smart Filters page, which will help you search for and edit projects and tasks.

Downloads

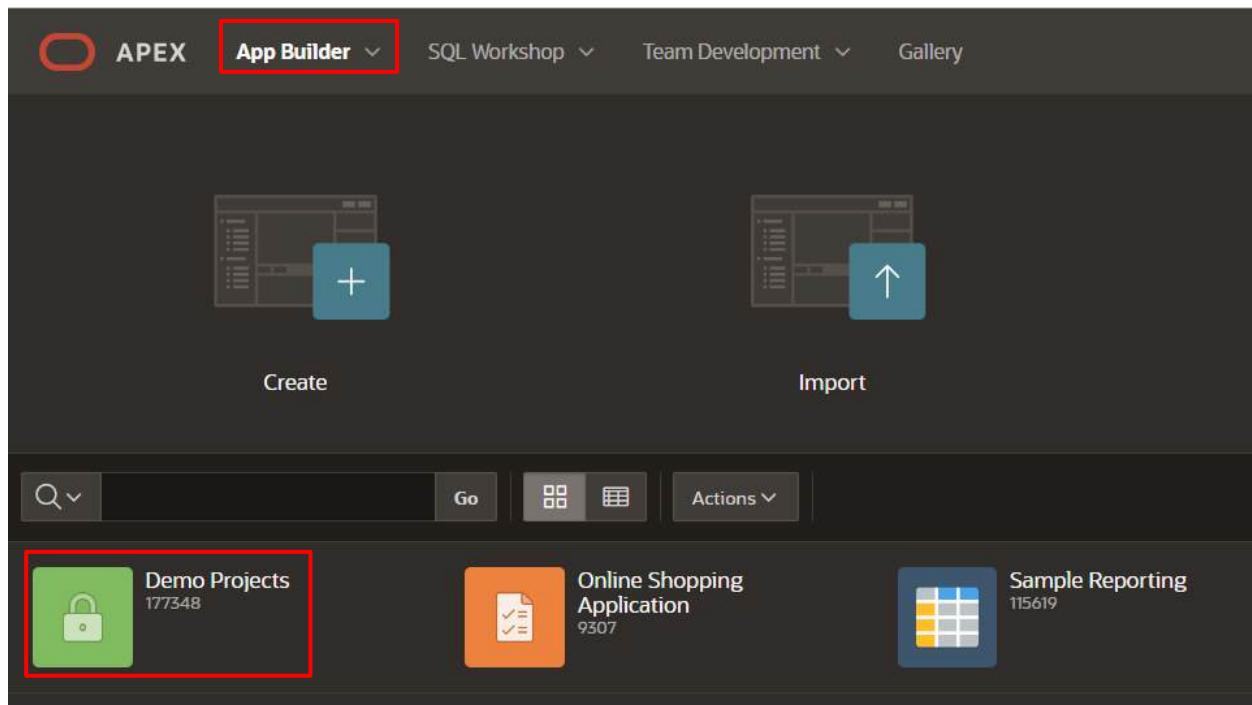
- Did you miss out on trying the previous practices? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, follow the steps described in the [Get Started with Oracle APEX](#) and [Using SOL Workshop](#) workshops.

Tasks

Add an Interactive Grid to the Demo Projects Application

In this practice, you create an Interactive Grid page on the **DEMO_PROJECTS** table in the **Demo Projects** application. You already created the Demo Projects application in HOL-3.

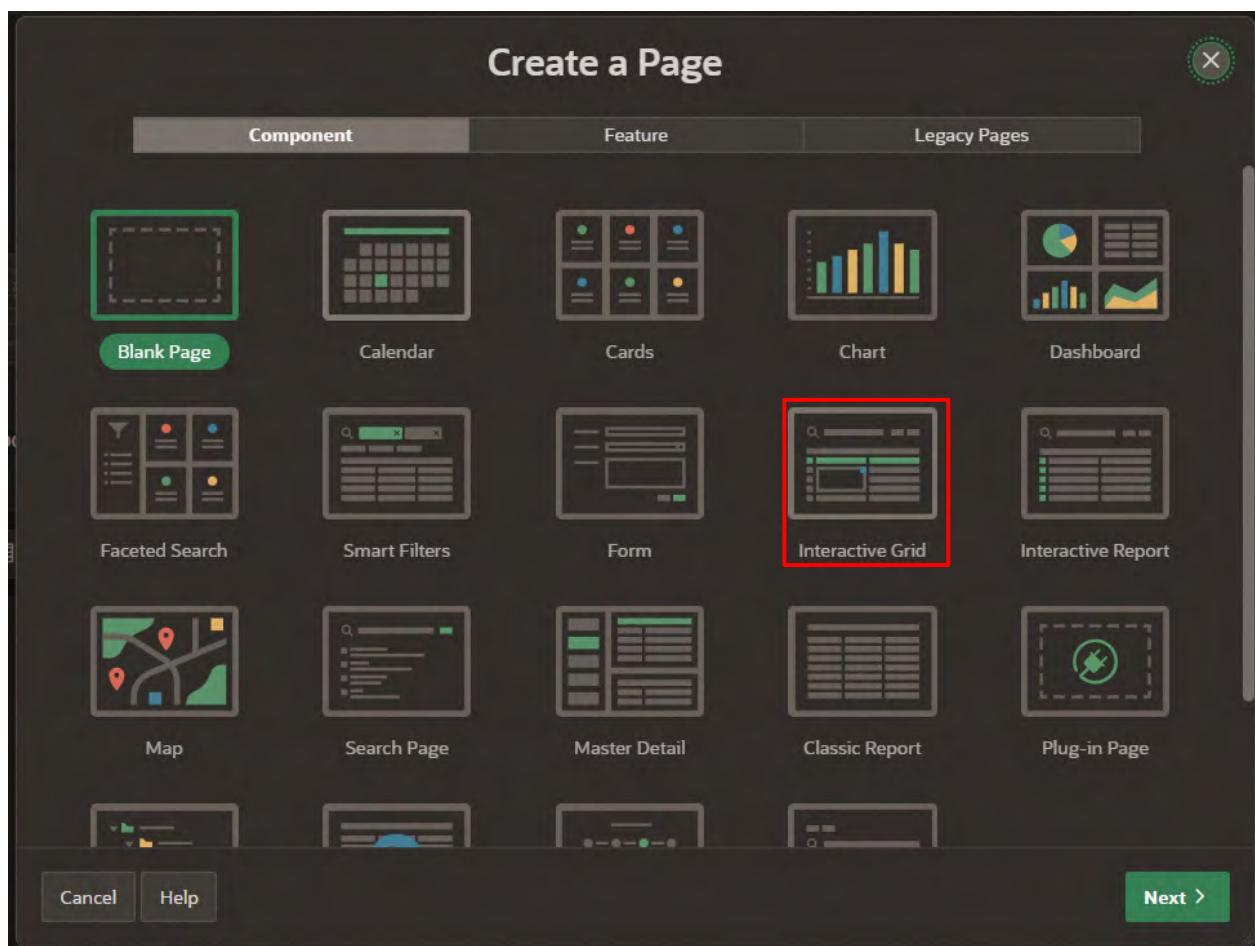
- On the Workspace home page, click **App Builder**. Select the **Demo Projects** application.



2. Click **Create Page**.



3. Under **Create a Page**, select **Component** and then select **Interactive Grid**.



4. For Page Attributes, enter/select the following:

Under **Page Definition**:

- Page Number: **4**
- Page Name: **Project Tasks**

Under **Data Source**:

- Table/View Name: **DEMO_PROJECTS**

Under **Navigation**:

- Use Breadcrumb: **No**

Click **Create Page**.

Create Interactive Grid

Page Definition

* Page Number (?)

* Name (?)

Page Mode Normal Modal Dialog Drawer (?)

Include Form Page On (?)

Data Source

Data Source Local Database REST Enabled SQL Service REST Data Source (?)

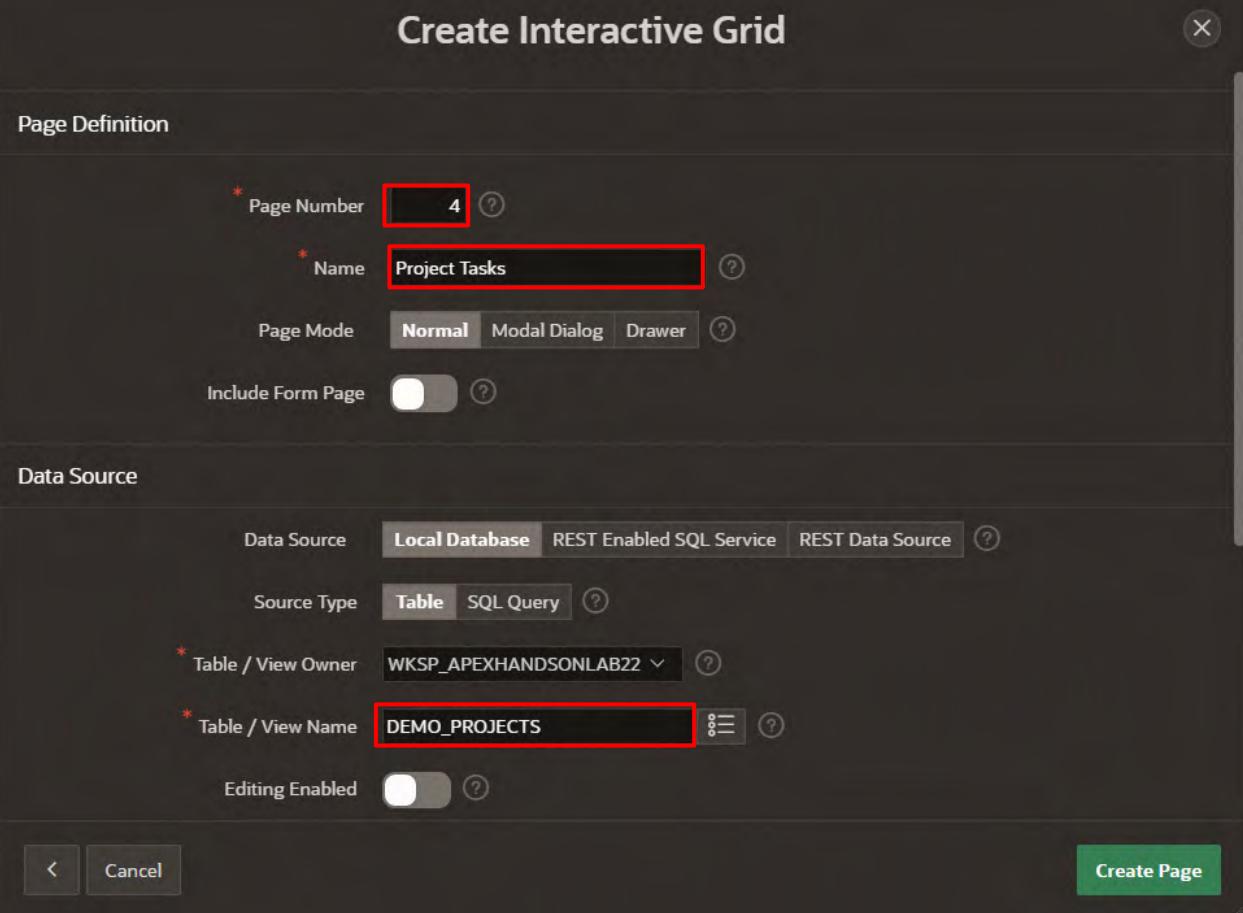
Source Type Table SQL Query (?)

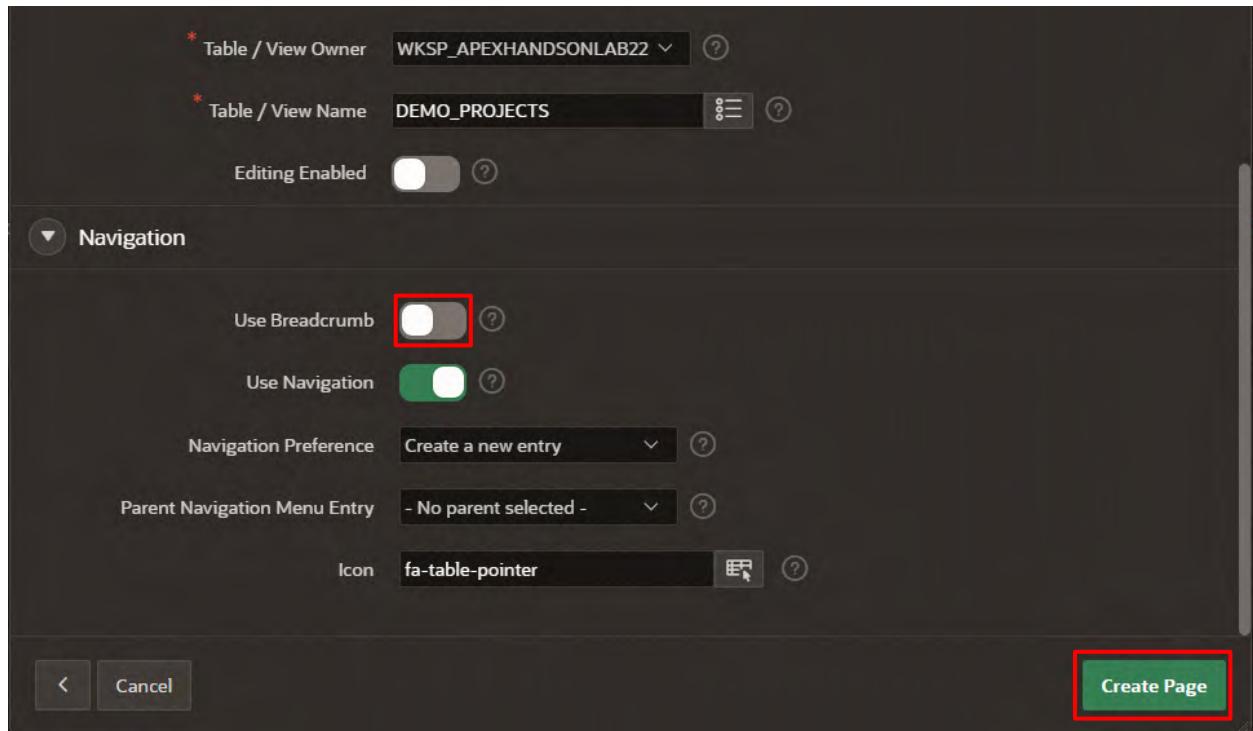
* Table / View Owner (?)

* Table / View Name (?)

Editing Enabled Off (?)

< Cancel Create Page





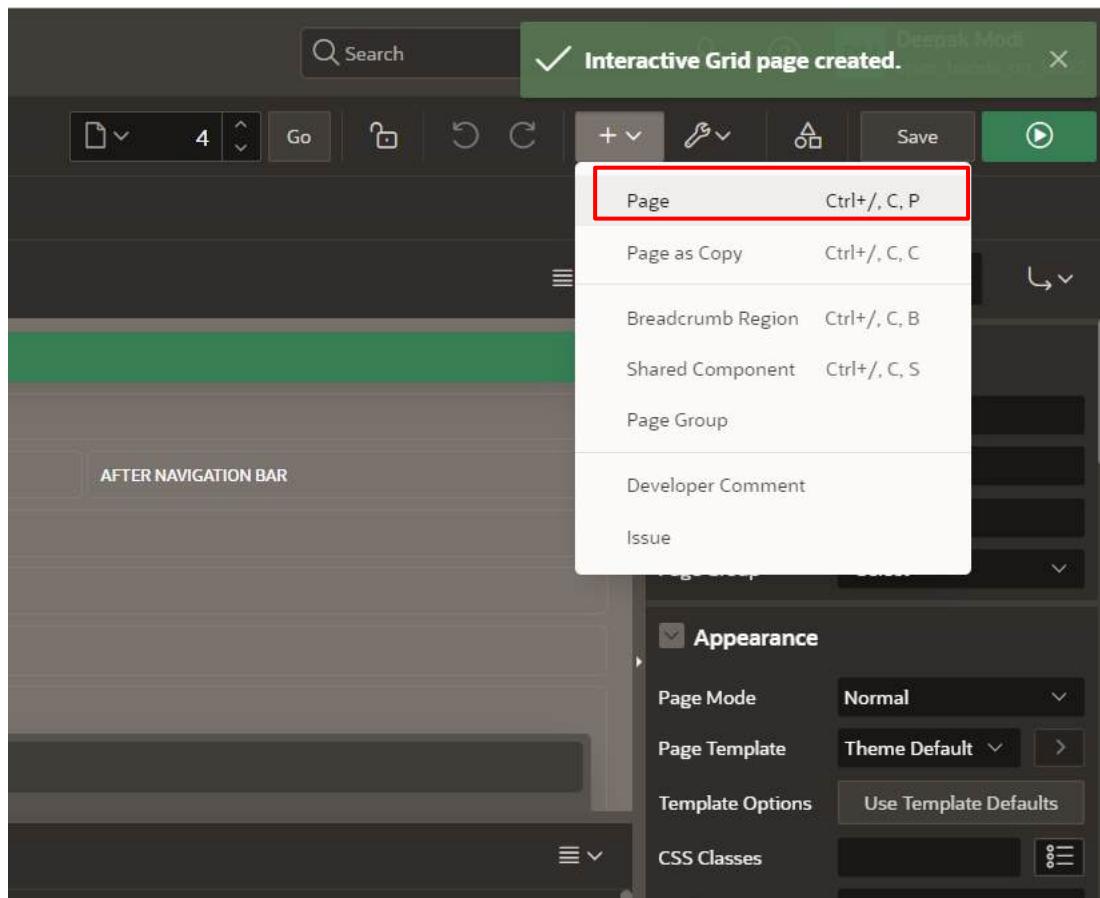
5. Now that you have created an **Interactive Grid** page, you can view it by clicking **Save** and **Run Page** on the top right.

| Demo Projects | | | | | | | | |
|----------------------------|--|--|------------|------------|---------|---------------|------|--------|
| Project | | Task Name | Start Date | End Date | Status | Assigned To | Cost | Budget |
| ACME Web Configuration | | Identify server requirements | 11/19/2021 | 12/6/2021 | Closed | John Watson | 300 | 300 |
| Maintain Support Systems | | HR software upgrades | 12/4/2021 | 1/1/2022 | On-Hold | Pam King | 8000 | 7000 |
| Maintain Support Systems | | Apply Billing System updates | 1/7/2022 | 1/29/2022 | On-Hold | Russ Sanders | 9500 | 7000 |
| ACME Web Configuration | | Determine Web listener configuration(s) | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| ACME Web Configuration | | Specify security authentication scheme(s) | 1/2/2022 | 1/4/2022 | Closed | John Watson | 200 | 300 |
| ACME Web Configuration | | Select servers for Development, Test, Production | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| Email Integration | | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3000 | 1500 |
| ACME Web Configuration | | Configure Workspace provisioning | 12/20/2021 | 1/4/2022 | Closed | John Watson | 200 | 100 |
| ACME Web Configuration | | Create pilot workspace | 1/15/2022 | 2/8/2022 | Closed | John Watson | 100 | 100 |
| ACME Web Configuration | | Run installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| Bug Tracker | | Implement bug tracking software | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| Bug Tracker | | Review automated testing tools | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff | 2750 | 1500 |
| Train Developers | | Publish development standards | 12/3/2021 | 12/13/2021 | On-Hold | John Watson | 1000 | 2000 |
| Train Developers | | Publish links to self-study courses | 12/28/2021 | 1/3/2022 | Closed | John Watson | 100 | 100 |
| Train Developers | | Create training workspace | 12/9/2021 | 12/11/2021 | Closed | James Cassidy | 500 | 700 |
| Load Packaged Applications | | Identify point solutions required | 1/2/2022 | 1/24/2022 | Closed | John Watson | 200 | 300 |

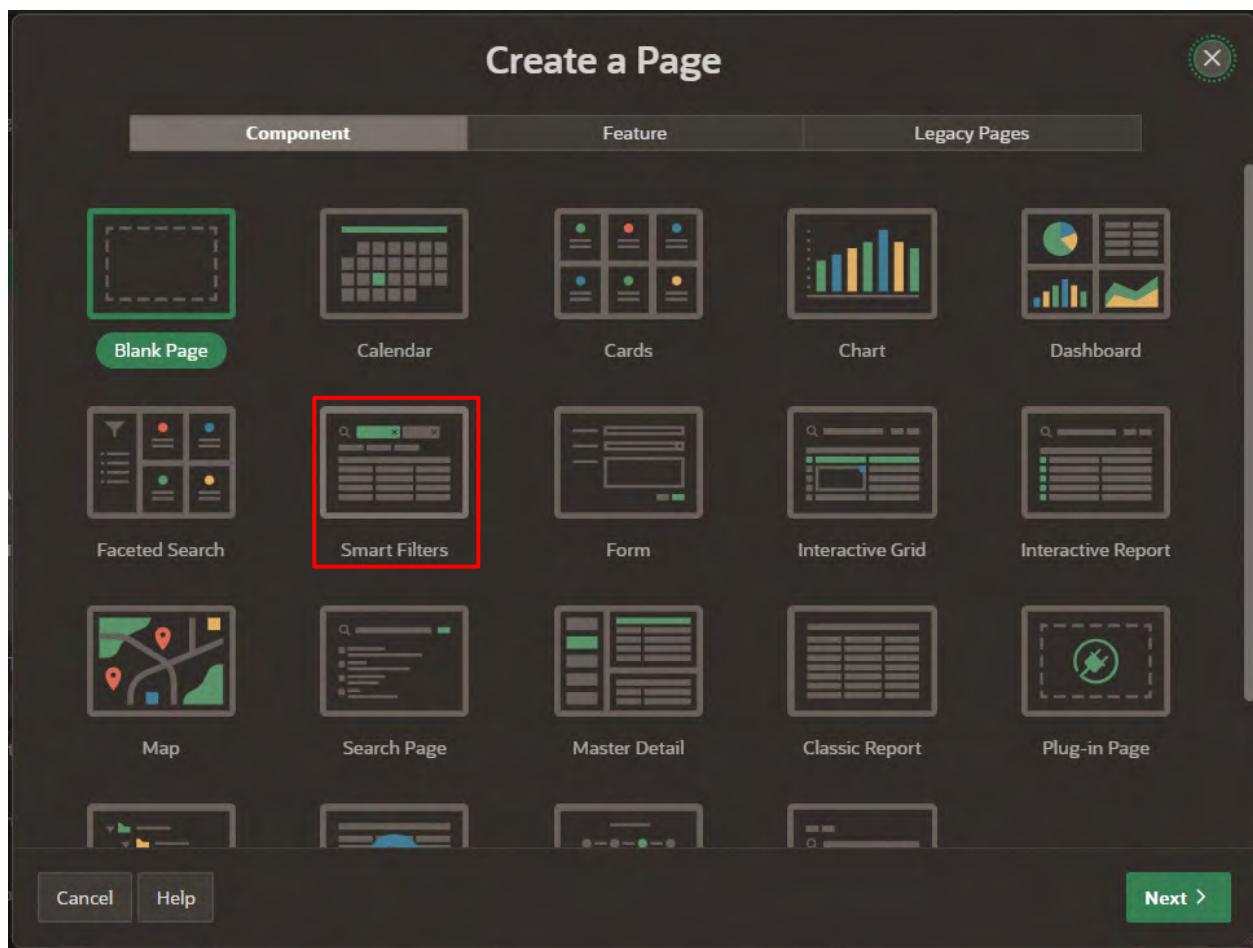
Add Smart Filters to the Demo Projects Application

In this practice, you create a Smart Filters report on the **DEMO_PROJECTS** table in the **Demo Projects** application. You already created the Demo Projects application in HOL-3.

1. Navigate to the Create (+) button and click **Page**.



2. Under **Create a Page**, select **Component** and then select **Smart Filters**.



3. For **Create Smart Filters**, enter/select the following:

Under Page Definition:

- Page Number: **8**
- Name: **Project Tasks Search**

Under Data Source:

- Table/View Name: **DEMO_PROJECTS**

Under Navigation:

- Use Breadcrumb: **No**

Click **Next**.

Create Smart Filters

Page Definition

* Page Number (?)

* Name (?)

Data Source

Data Source Local Database REST Enabled SQL Service REST Data Source (?)

Source Type Table SQL Query (?)

* Table / View Owner (?)

* Table / View Name (?)

Navigation

Use Breadcrumb (?)

Use Navigation (?)

Navigation Preference (?)

Parent Navigation Menu Entry (?)

< Cancel Next >

4. On the **Select the Filters** page:

- Leave the remaining details to the defaults
- Click **Create Page**

Create Smart Filters

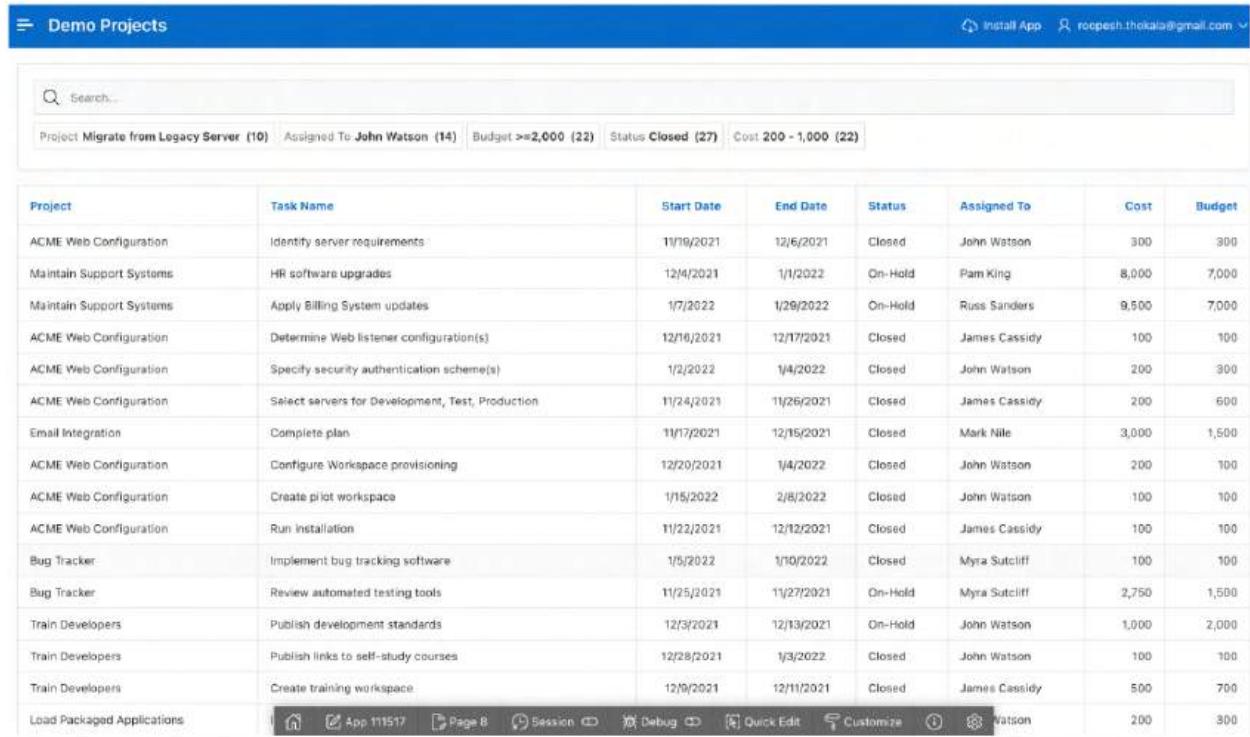
Display as **Report** Cards [?](#)

Select the Filter you want to include in this report. Note that Filter recommendations are based on statistics of data in your table. You can refresh the column statistics to provide better recommendations. [Refresh](#)

| | Column | Filter | Searchable | Nulls | Distinct | Average Length |
|-------------------------------------|------------------------|----------|------------|-------|----------|----------------|
| | ID (number) | | No | 0 | 70 | 3 |
| <input checked="" type="checkbox"/> | PROJECT (varchar2) | Checkbox | Yes | 0 | 14 | 23 |
| <input type="checkbox"/> | TASK_NAME (varchar2) | Checkbox | Yes | 0 | 65 | 28 |
| | START_DATE (date) | | No | 0 | 51 | 8 |
| | END_DATE (date) | | No | 0 | 46 | 8 |
| <input checked="" type="checkbox"/> | STATUS (varchar2) | Checkbox | Yes | 0 | 4 | 7 |
| <input checked="" type="checkbox"/> | ASSIGNED_TO (varchar2) | Checkbox | Yes | 0 | 13 | 12 |
| <input checked="" type="checkbox"/> | COST (number) | Range | No | 0 | 22 | 3 |
| <input checked="" type="checkbox"/> | BUDGET (number) | Range | No | 0 | 19 | 4 |

< Cancel **Create Page**

5. Now that you have created a **Smart Filters** page, you can view it by clicking **Save** and **Run Page**.



The screenshot shows a grid-based application interface titled "Demo Projects". At the top, there is a search bar and several filter buttons: "Project Migrate from Legacy Server (10)", "Assigned To John Watson (14)", "Budget >=2,000 (22)", "Status Closed (27)", and "Cost 200 - 1,000 (22)". The main grid displays 18 rows of task data, each with columns for Project, Task Name, Start Date, End Date, Status, Assigned To, Cost, and Budget. The data includes tasks like "Identify server requirements" for ACME Web Configuration and "Review automated testing tools" for Bug Tracker. The bottom of the grid has a toolbar with icons for refresh, export, session, debug, quick edit, customize, and Watson.

| Project | Task Name | Start Date | End Date | Status | Assigned To | Cost | Budget |
|----------------------------|--|------------|------------|---------|---------------|-------|--------|
| ACME Web Configuration | Identify server requirements | 11/19/2021 | 12/6/2021 | Closed | John Watson | 300 | 300 |
| Maintain Support Systems | HR software upgrades | 12/4/2021 | 1/1/2022 | On-Hold | Pam King | 8,000 | 7,000 |
| Maintain Support Systems | Apply Billing System updates | 1/7/2022 | 1/29/2022 | On-Hold | Russ Sanders | 9,500 | 7,000 |
| ACME Web Configuration | Determine Web listener configuration(s) | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| ACME Web Configuration | Specify security authentication scheme(s) | 1/2/2022 | 1/4/2022 | Closed | John Watson | 200 | 300 |
| ACME Web Configuration | Select servers for Development, Test, Production | 1/14/2021 | 1/26/2021 | Closed | James Cassidy | 200 | 600 |
| Email Integration | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3,000 | 1,500 |
| ACME Web Configuration | Configure Workspace provisioning | 12/20/2021 | 1/4/2022 | Closed | John Watson | 200 | 100 |
| ACME Web Configuration | Create pilot workspace | 1/15/2022 | 2/8/2022 | Closed | John Watson | 100 | 100 |
| ACME Web Configuration | Run installation | 1/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| Bug Tracker | Implement bug tracking software | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| Bug Tracker | Review automated testing tools | 1/25/2021 | 1/27/2021 | On-Hold | Myra Sutcliff | 2,750 | 1,500 |
| Train Developers | Publish development standards | 12/3/2021 | 12/13/2021 | On-Hold | John Watson | 1,000 | 2,000 |
| Train Developers | Publish links to self-study courses | 12/28/2021 | 1/3/2022 | Closed | John Watson | 100 | 100 |
| Train Developers | Create training workspace | 12/9/2021 | 12/11/2021 | Closed | James Cassidy | 500 | 700 |
| Load Packaged Applications | | | | | | | |

You now know how to create an Interactive Grid page and a Smart Filters page. You may now **proceed to the next practice**.

Practice 2: Create Cards Regions and Faceted Search

Overview

In this practice, you create:

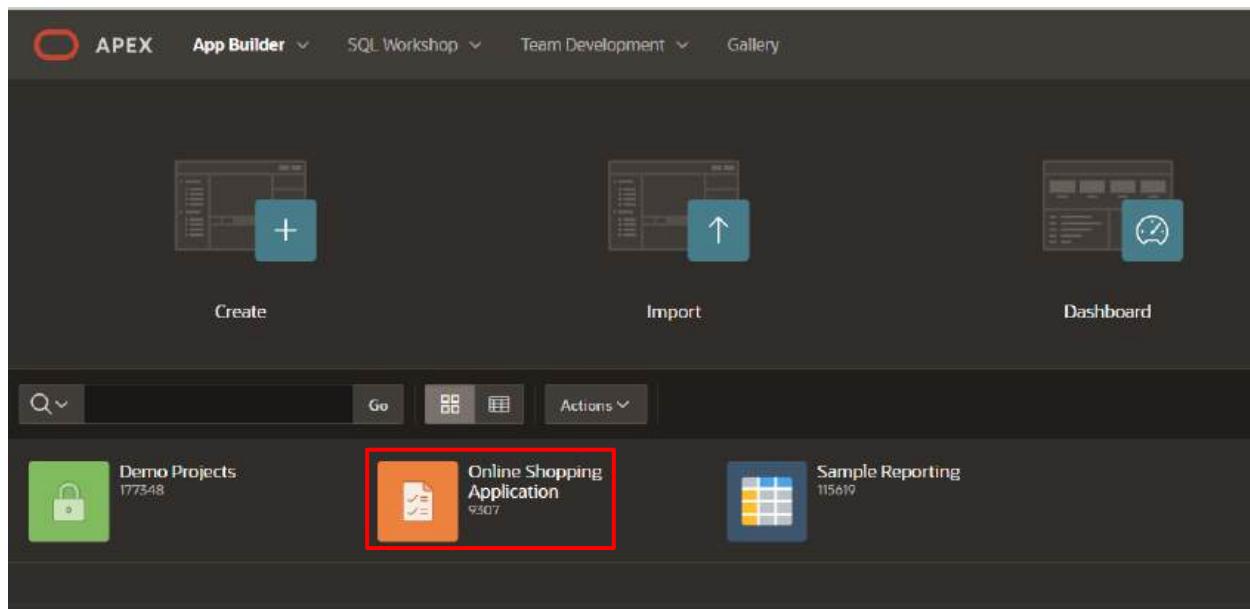
- **Application Items** and an **Application Process**
- A **Cards** page to view **Product Details** and **Customer Reviews** in the online Shopping Cart application
- A **Faceted Search** page to view **Products**

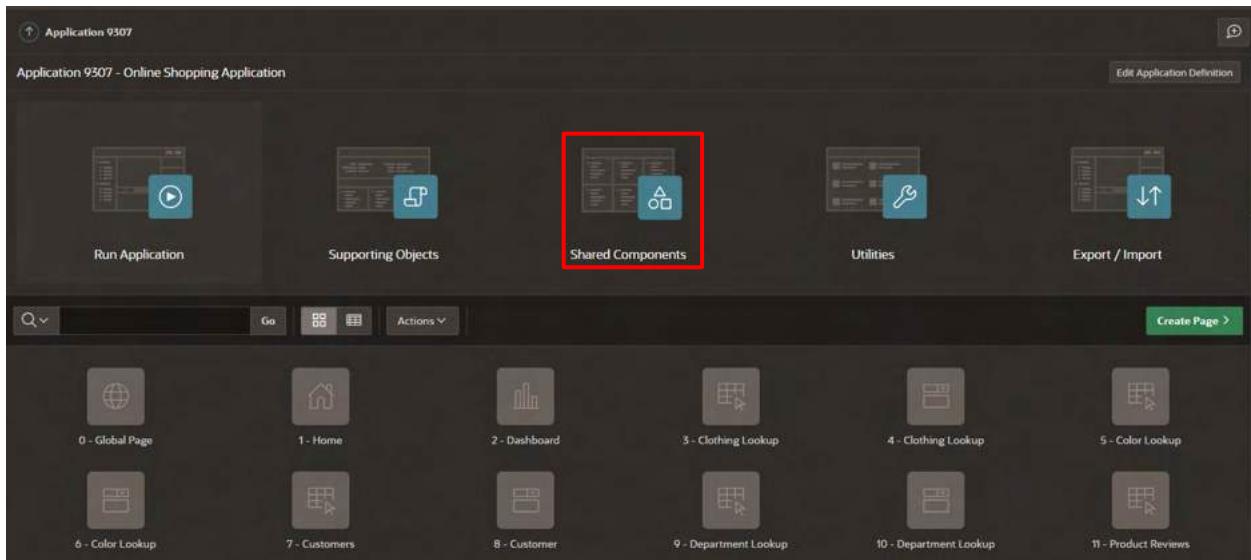
Tasks

Create Application Items

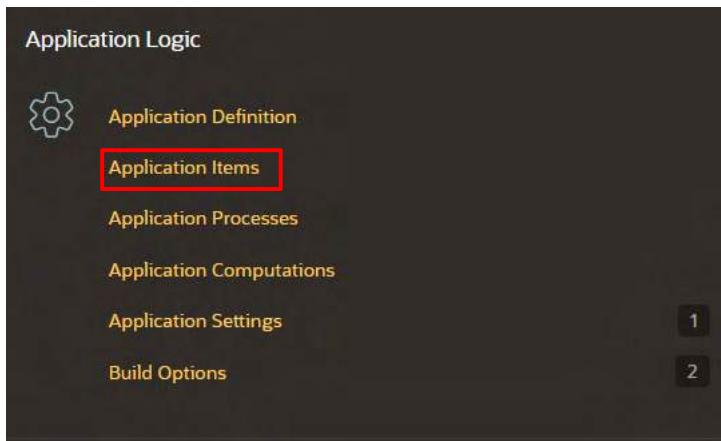
This is needed to count the number of items in the shopping cart and the icon to display in the navigation bar.

1. Navigate to **Online Shopping Application** and click **Shared Components**.

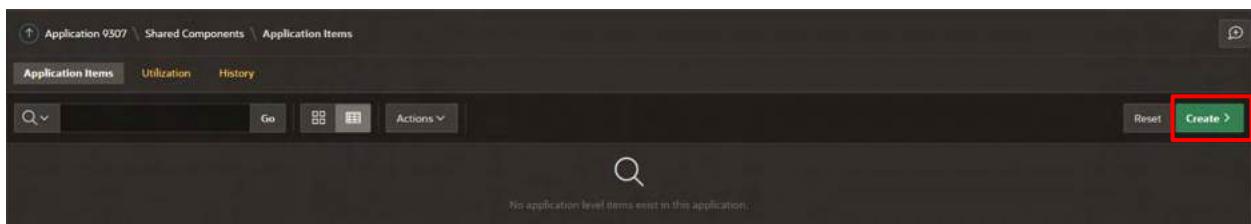




2. Under **Application Logic**, click **Application Items**.



3. Click **Create**.

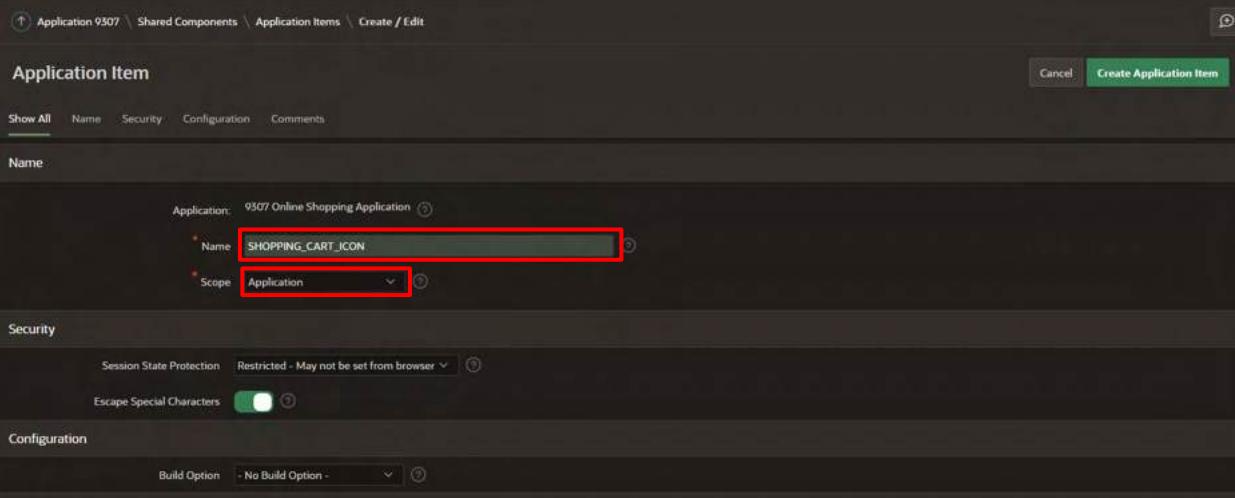


4. Create two items as follows:

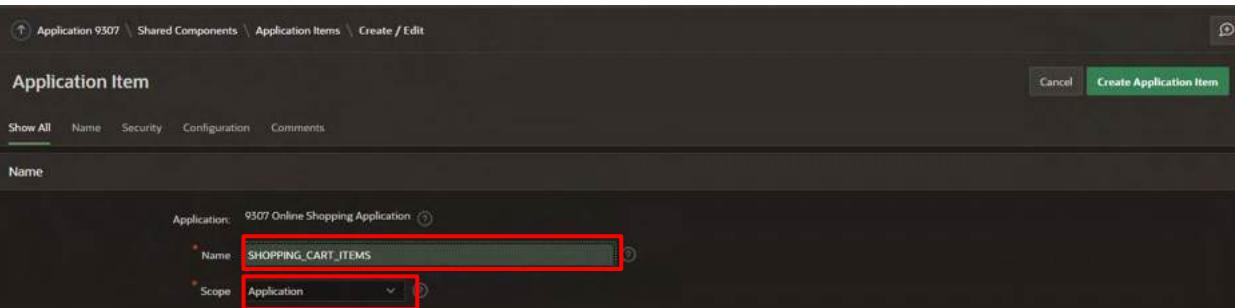
Table 1: Develop Reports | Practice 2: Create Cards Regions and Faceted Search

| Name | Scope |
|---------------------|-------------|
| SHOPPING_CART_ICON | Application |
| SHOPPING_CART_ITEMS | Application |

5. Click **Create Application Item** and create the second item.



The screenshot shows the 'Create Application Item' dialog for the 'SHOPPING_CART_ICON' item. The 'Name' field is set to 'SHOPPING_CART_ICON' and the 'Scope' is set to 'Application'. Both fields are highlighted with a red box. The 'Security' section includes 'Session State Protection' (set to 'Restricted - May not be set from browser') and 'Escape Special Characters' (checkbox checked). The 'Configuration' section shows 'Build Option' as 'No Build Option'. The 'Comments' section is empty.



The screenshot shows the 'Create Application Item' dialog for the 'SHOPPING_CART_ITEMS' item. The 'Name' field is set to 'SHOPPING_CART_ITEMS' and the 'Scope' is set to 'Application'. Both fields are highlighted with a red box. The 'Security' section includes 'Session State Protection' (set to 'Restricted - May not be set from browser') and 'Escape Special Characters' (checkbox checked). The 'Configuration' section shows 'Build Option' as 'No Build Option'. The 'Comments' section is empty.

Create the Application Process

This process is needed to refresh the number of items in the shopping cart, which will be shown in the navigation bar.

1. Click **Shared Components**.

The screenshot shows the Oracle APEX App Builder interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below the navigation is a search bar and a toolbar with icons for back, forward, and help. The main content area has a breadcrumb path: Application 9307 \ Shared Components. A success message 'Item processed.' is displayed. The page title is 'Application Items'. There are tabs for Application Items, Utilization, and History. A search bar and filter buttons are at the top. A table lists application items with columns: Name, Computed On, Updated, Updated By, Protection Level, Escape Special Characters, and Scope. Two items are listed: SHOPPING_CART_ICON and SHOPPING_CART_ITEMS.

| Name | Computed On | Updated | Updated By | Protection Level | Escape Special Characters | Scope |
|---------------------|-------------|----------------|--------------------------|--|---------------------------|-------------|
| SHOPPING_CART_ICON | - | 70 seconds ago | MODIDEEPAK1991@GMAIL.COM | Restricted - May not be set from browser | Yes | Application |
| SHOPPING_CART_ITEMS | - | 2 seconds ago | MODIDEEPAK1991@GMAIL.COM | Restricted - May not be set from browser | Yes | Application |

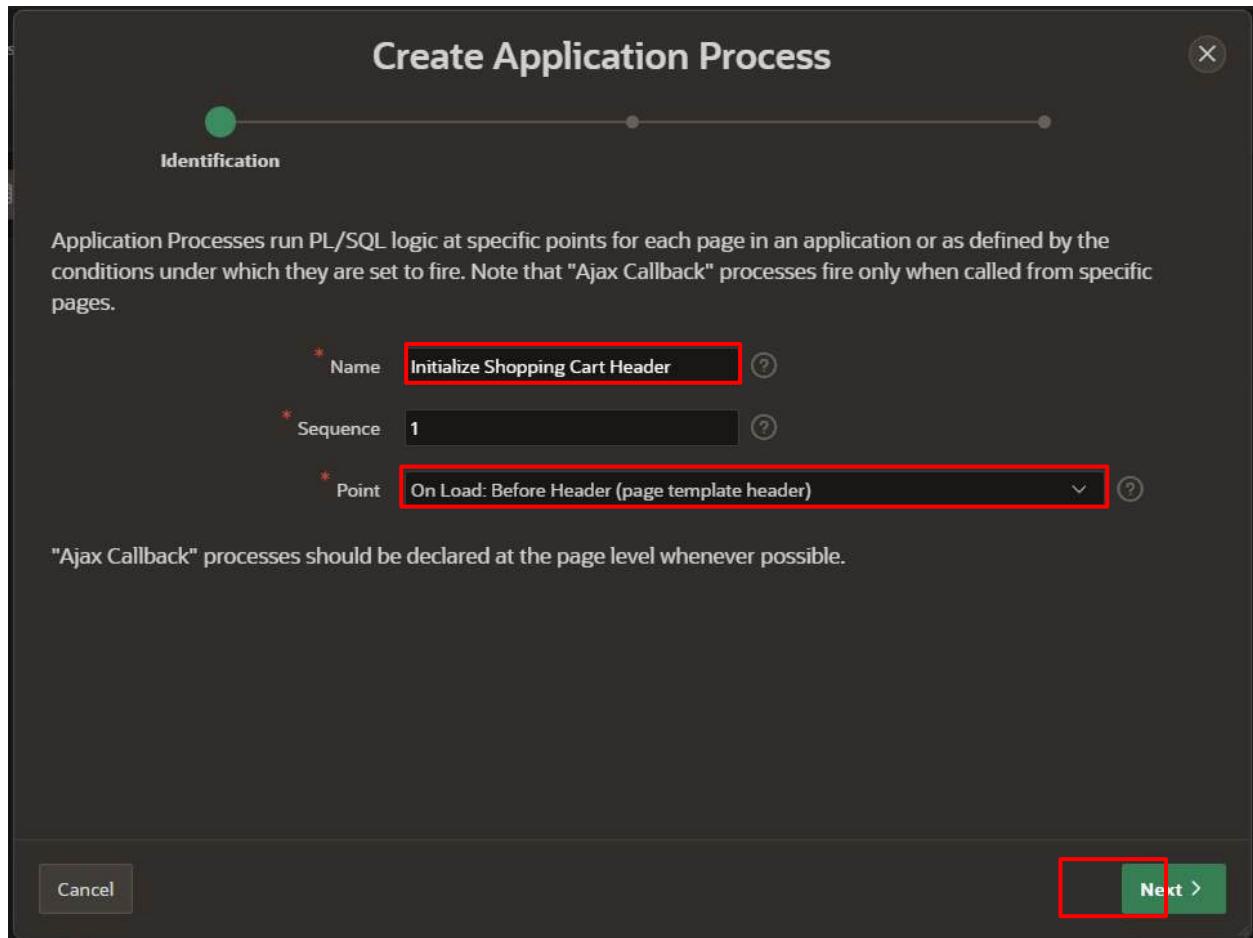
2. Under **Application Logic**, click **Application Processes**.

The screenshot shows the Oracle APEX App Builder interface. The top navigation bar is the same as the previous screenshot. The main content area has a breadcrumb path: Application 9307 \ Shared Components. On the left, under 'Application Logic', there is a sidebar with icons and labels: Application Definition (selected), Application Items (with a count of 2), Application Processes (selected and highlighted with a red box), Application Computations, Application Settings (with a count of 1), and Build Options (with a count of 2). On the right, under 'Security', there are links for Security Attributes, Authentication Schemes, Authorization Schemes, Application Access Control, and Session State Protection.

3. Click **Create** and enter/select the following:

- Name: **Initialize Shopping Cart Header**
- Process Point: **On Load: Before Header (page template header)**

The screenshot shows the Oracle APEX App Builder interface. The top navigation bar is the same. The main content area has a breadcrumb path: Application 9307 \ Shared Components \ Application Processes. The page title is 'Application Processes'. A toolbar at the top includes a search bar, filter buttons, and a 'Create' button (which is highlighted with a red box). The main area displays a message 'No data found.'



4. Click **Next**.

5. For **Code**, enter:

```
-- Initialize shopping cart navigation bar to show appropriate
icon and count
DECLARE
    l_cnt NUMBER := manage_orders.get_quantity;
BEGIN
    IF l_cnt > 0 THEN
        :SHOPPING_CART_ITEMS := l_cnt;
        :SHOPPING_CART_ICON := 'fa-cart-full';
    ELSE
        :SHOPPING_CART_ITEMS := '';
        :SHOPPING_CART_ICON := 'fa-cart-empty';
    END IF;
END;
```

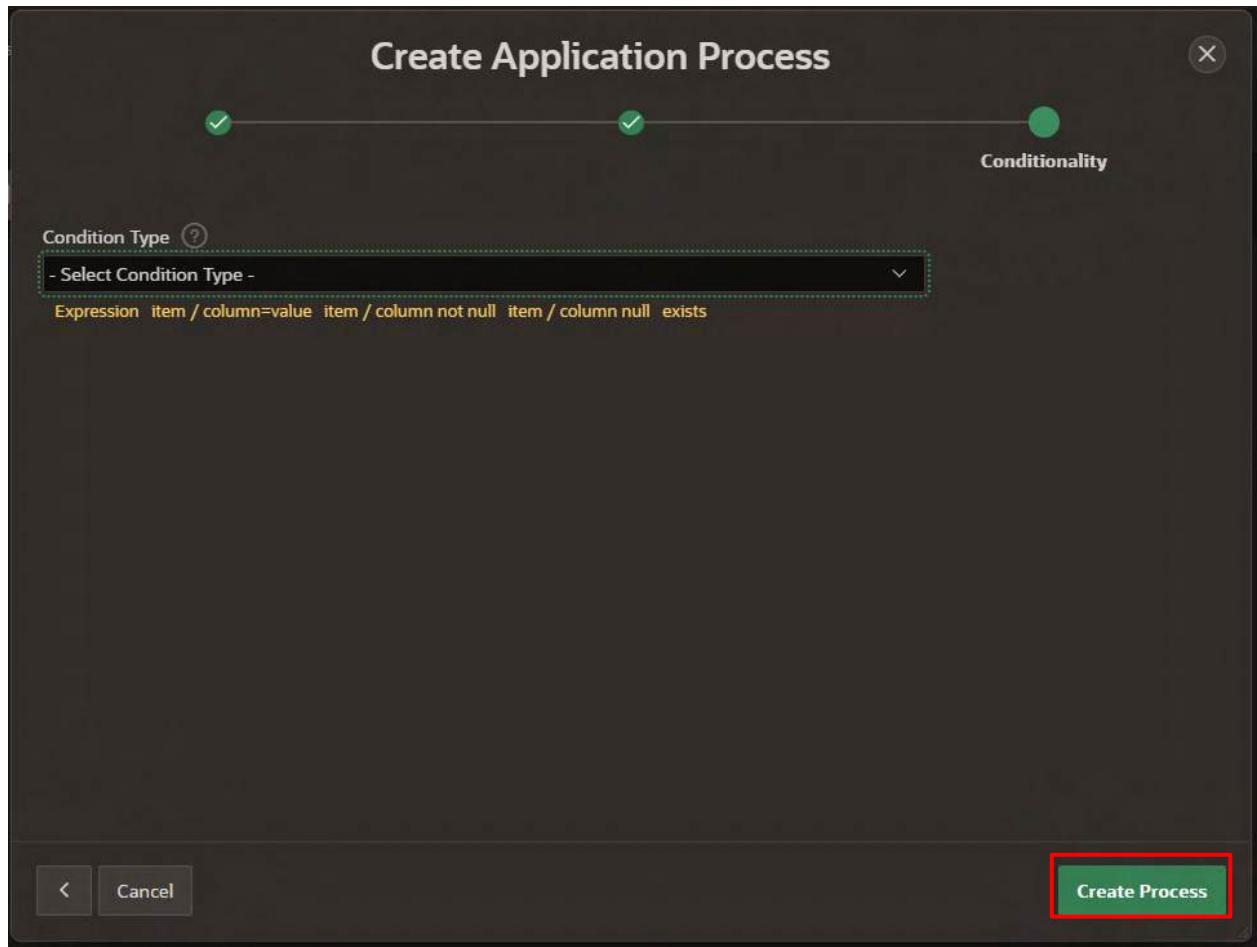
6. Click **Next**.

The screenshot shows the 'Create Application Process' dialog in Oracle Application Express. The title bar says 'Create Application Process'. Below it is a progress bar with three steps: the first is green with a checkmark, the second is grey labeled 'Source', and the third is grey. The language is set to 'PL/SQL'. The code editor contains the following PL/SQL code:

```
1 -- Initialize shopping cart navigation bar to show appropriate icon and count
2 DECLARE
3     l_cnt NUMBER := manage_orders.get_quantity;
4 BEGIN
5     IF l_cnt > 0 THEN
6         :SHOPPING_CART_ITEMS := l_cnt;
7         :SHOPPING_CART_ICON := 'fa-cart-full';
8     ELSE
9         :SHOPPING_CART_ITEMS := '';
10        :SHOPPING_CART_ICON := 'fa-cart-empty';
11    END IF;
12 END;
```

Below the code editor is a checkbox labeled 'Do not validate code (parse code at runtime only)'. At the bottom are 'Cancel' and 'Next >' buttons.

7. Click **Create Process**.



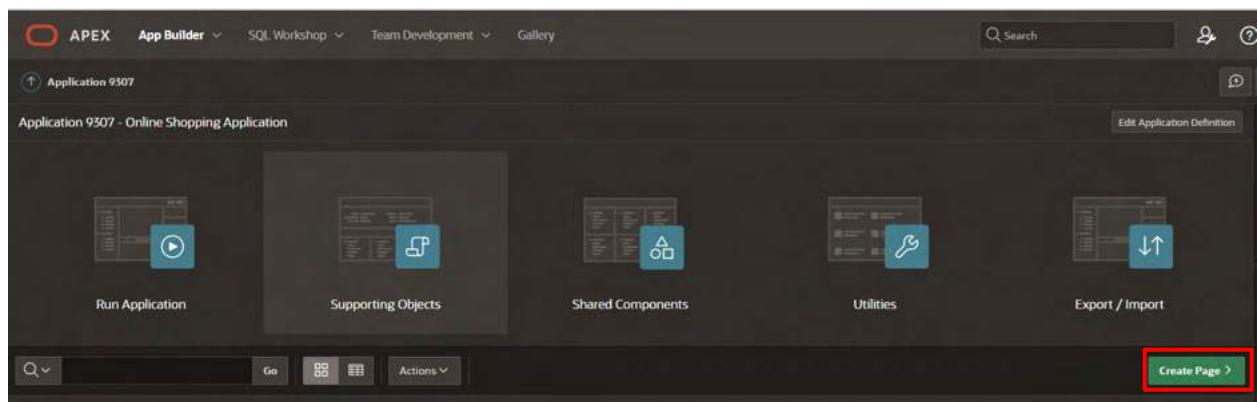
Create a Normal Page - Shopping Cart

The shopping cart page allows users to review and edit the products in the cart. Users can also create an order or clear the shopping cart.

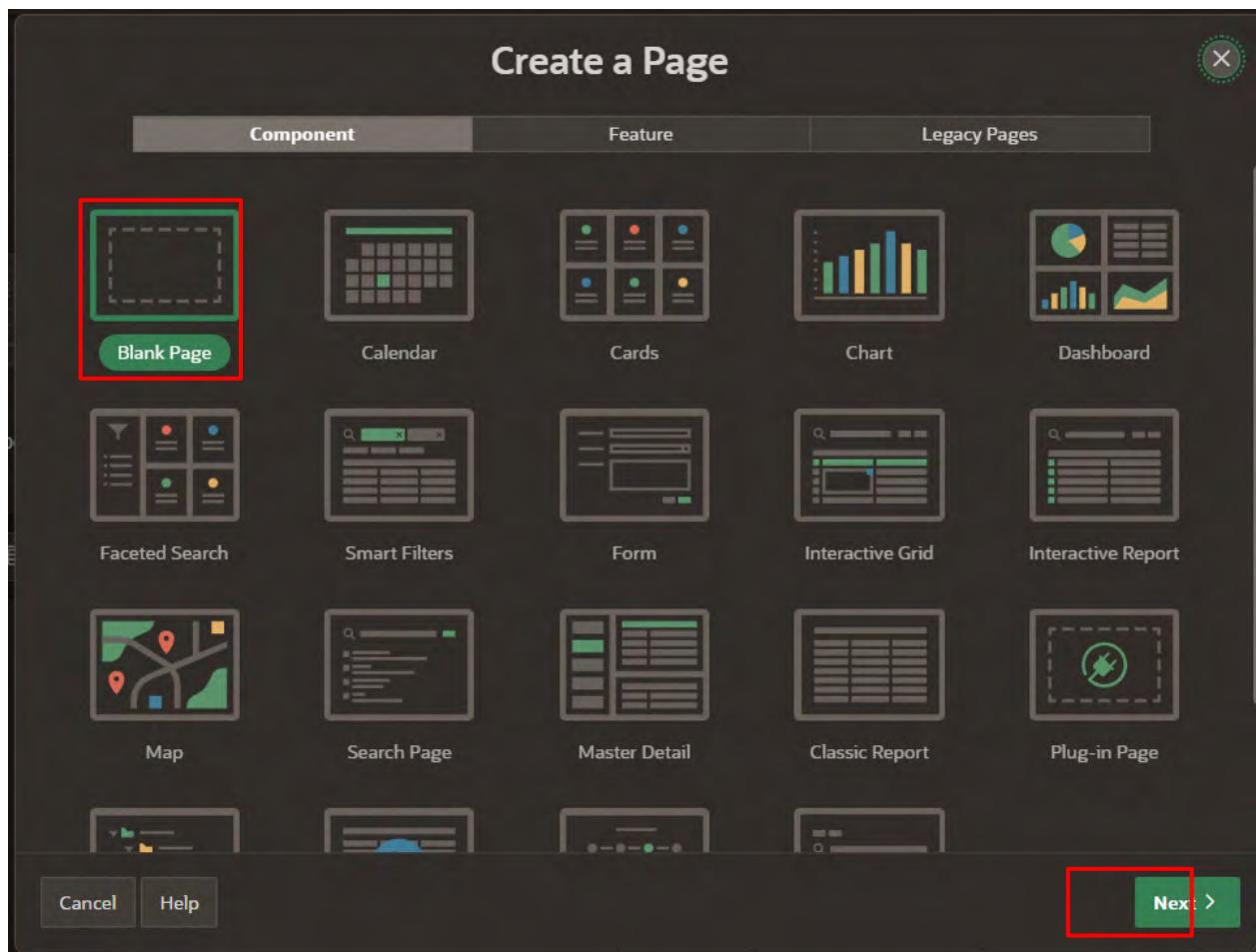
1. Click **Application Home**. *The ID of your application may vary.*

| Sequence | Name | Point | Actions |
|----------|---------------------------------|---|---|
| 1 | Initialize Shopping Cart Header | On Load: Before Header (page template header) | -- Initialize shopping cart navigation bar to show appropriate icon an... I_cnt > 0 THEN :SHOPPING_CART_ITEMS := 1 ... |

2. Click **Create Page**.



3. Select **Blank Page** and click **Next**.



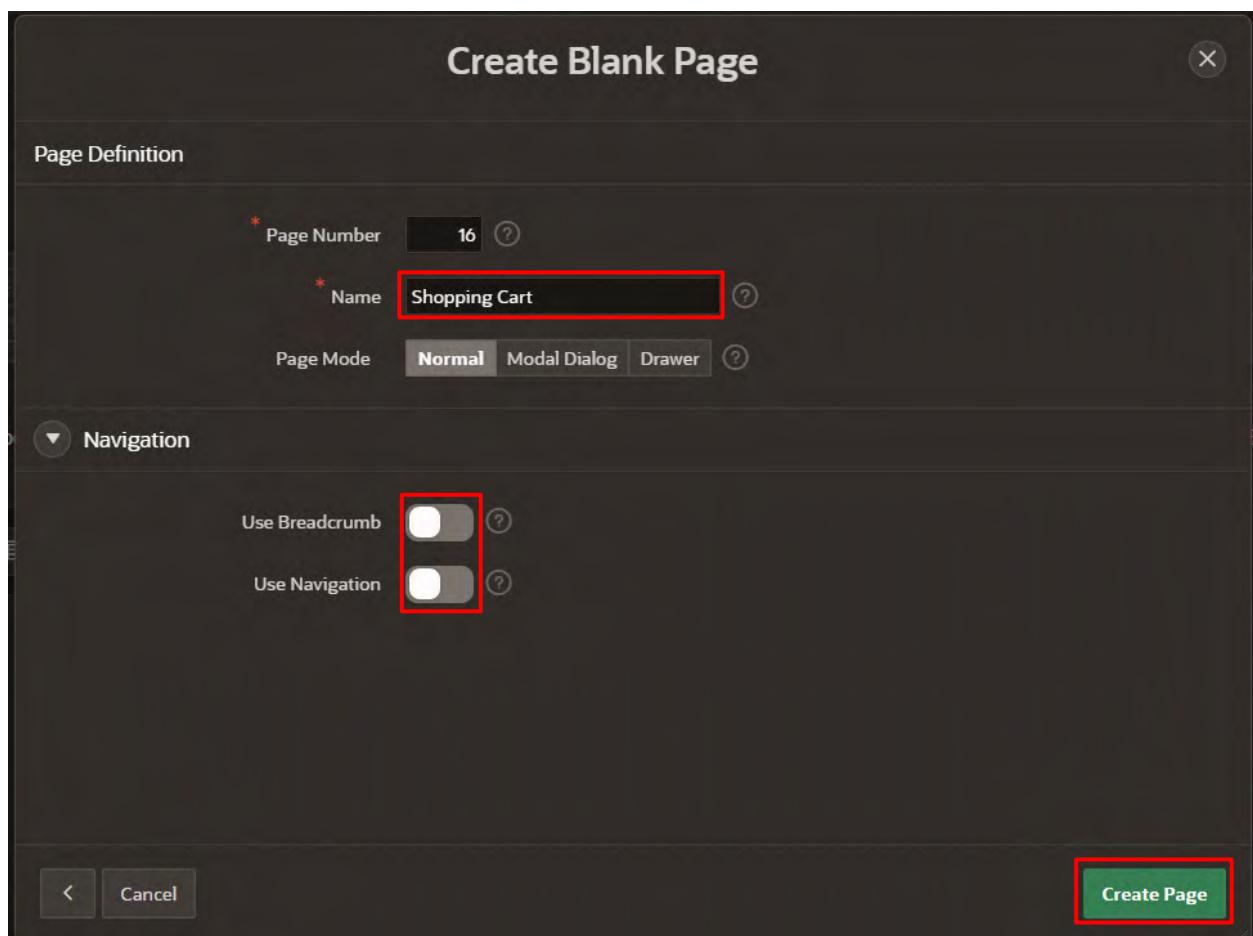
4. For **Create Blank Page**, enter/select the following:

Under Page Definition:

- Page Number: **16**
- Name: **Shopping Cart**

Under Navigation:

- Use Breadcrumb: **No**
- Use Navigation: **No**

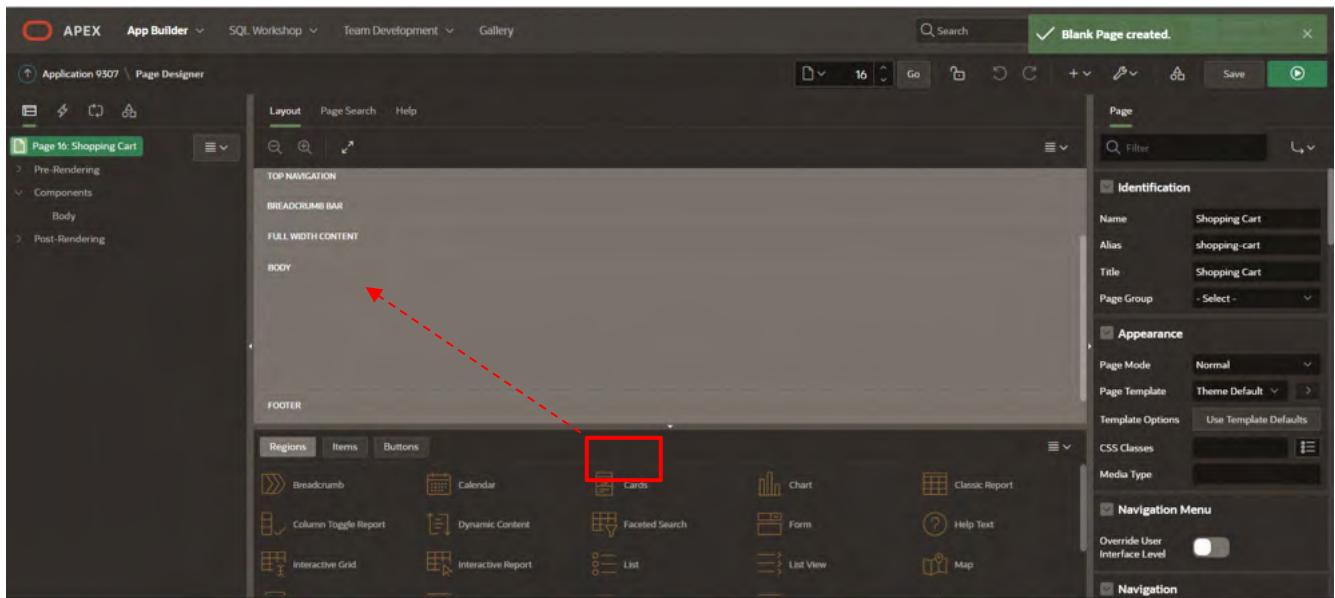


Add a Cards Region

This region will list the items that have been added temporarily to the shopping cart.

1. On the new page created, navigate to the **Gallery Menu**.

2. Drag a **Cards** region and drop it into the **Body** section.



3. In the Property Editor, enter/select the following:

Under Identification:

- Title: **Shopping Cart**

Under Source:

- Type: **SQL Query**
- SQL Query:

```
SELECT seq_id item,
       p.product_image,
       p.product_id,
       p.product_name name,
       p.unit_price,
       n002           quantity,
       p.unit_price* n002 subtotal,
       b.brand
  FROM apex_collections a,
       products p,
       json_table (p.product_details, '$' columns ( brand
varchar2(4000) path '$.brand') ) b
 WHERE collection_name = 'PRODUCTS'
   AND p.product_id = a.n001
```

The screenshot shows the 'Region' configuration page in Oracle APEX. The 'Identification' section has 'Title' set to 'Shopping Cart'. The 'Source' section has 'Location' set to 'Local Database' and 'Type' set to 'SQL Query'. The 'SQL Query' field contains the following code:

```
SELECT seq_id item,
       p.product_image,
       p.product_id,
       p.product_name name,
       p.unit_price,
       n002           quantity,
       p.unit_price* n002 subtotal,
       b.brand
  FROM apex_collections a,
       products p,
       json_table (p.product_details, '$'
      columns ( brand varchar2(4000) path
      '$.brand' ) ) b
 WHERE collection_name = 'PRODUCTS'
 AND   p.product_id = a.n001
```

4. Click **Attributes** and enter/select the following:

Under Appearance:

- Layout: **Horizontal (Row)**

Under Title:

- Column: **NAME**

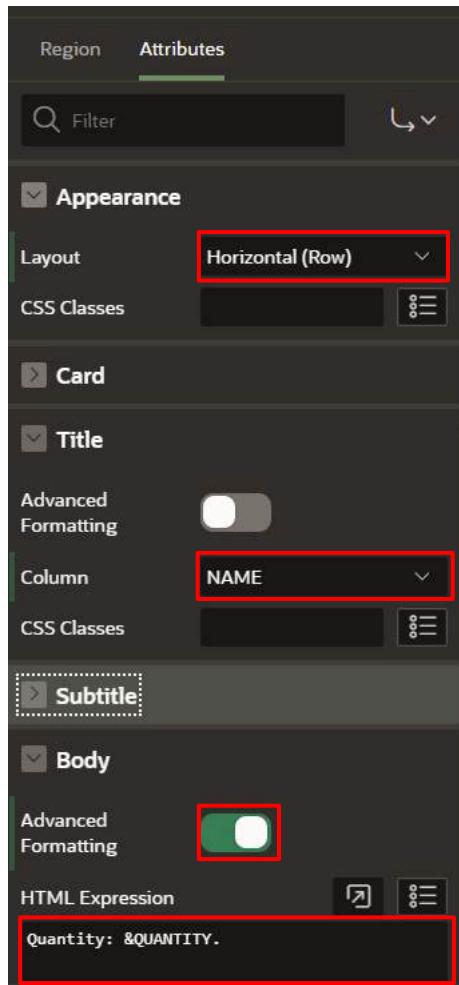
Under Subtitle:

- Column: **BRAND**

Under Body:

- Advanced Formatting: **On**
- HTML Expression:

```
Quantity: &QUANTITY.
```



Under Secondary Body:

- Advanced Formatting: **On**
- HTML Expression:

```
<b>Unit Price: &UNIT_PRICE. </b> <BR>
<b>Subtotal: &SUBTOTAL. </b>
```

Under Media:

- Source: **BLOB Column**
- BLOB Column: **PRODUCT_IMAGE**

The screenshot shows the Oracle Reports Attributes panel. At the top, there are tabs for Region and Attributes, with Attributes selected. Below the tabs is a search bar labeled "Filter". Under the "Attributes" tab, there are two main sections: "Secondary Body" and "Media".

Secondary Body:

- "Advanced Formatting" toggle switch is turned on.
- "HTML Expression" field contains the following code:

```
<b>Unit Price: &UNIT_PRICE. </b> <BR>
<b>Subtotal: &SUBTOTAL. </b>
```

This field is highlighted with a red rectangle.

Media:

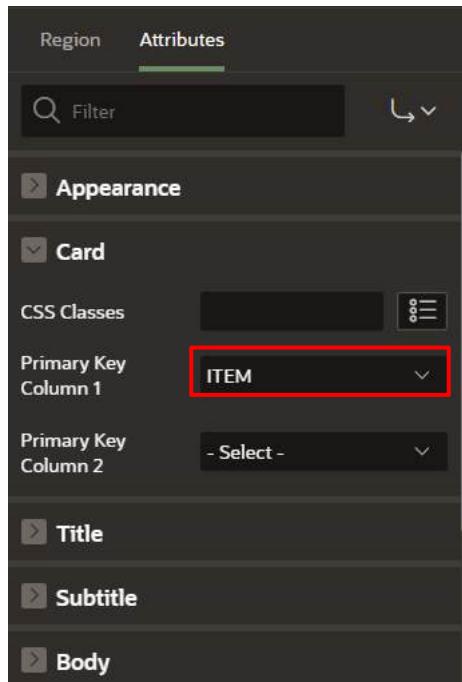
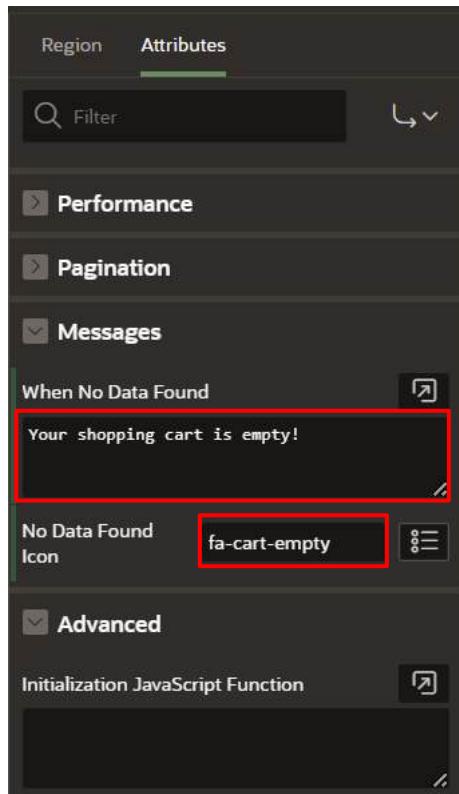
- "Advanced Formatting" toggle switch is turned off.
- "Source" dropdown is set to "BLOB Column".
- "BLOB Column" dropdown is set to "PRODUCT_IMAGE". This field is highlighted with a red rectangle.
- "Position" dropdown is set to "Body".
- "Appearance" dropdown is set to "Auto".
- "Sizing" dropdown is set to "Fit".
- "CSS Classes" and "Image Description" fields are empty.

Under **Messages**:

- When No Data Found: **Your shopping cart is empty!**
- For No Data Found Icon: **fa-cart-empty**

Under **Card**:

- Primary Key Column 1: **ITEM**

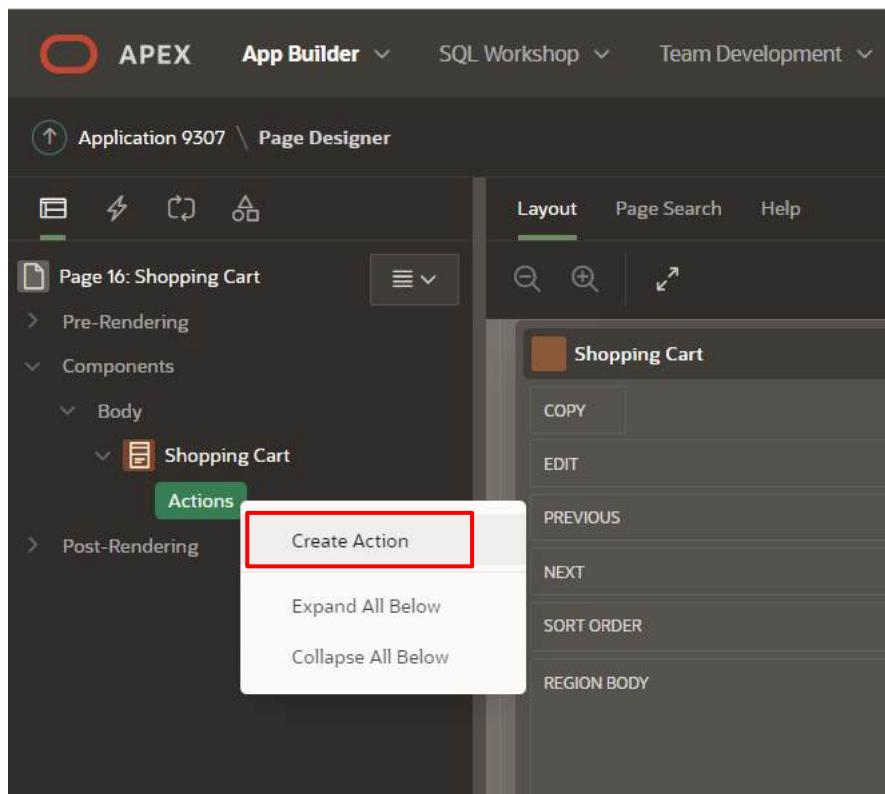


Add an Action to the Shopping Cart

This action allows customers to open a page to edit a particular item in the shopping cart.

1. In the Rendering tree (left pane), navigate to **Actions** under **Shopping Cart**.

- Right-click **Actions** and select **Create Action**.



- In the Property Editor:

- For Label, enter **Edit**.
- For Target, enter **No Link Defined**.
- For Page, enter **17**.

{Note: Page 17 will be created in the next practice}

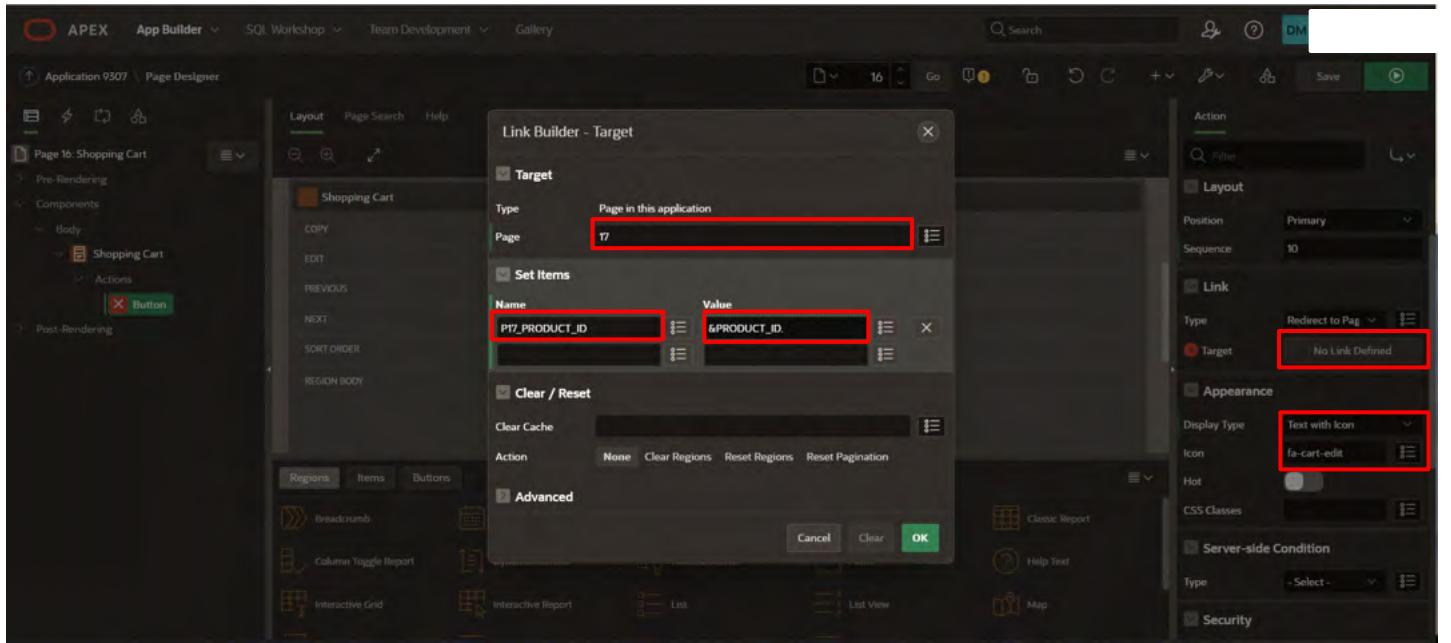
- Set items as follows:

Table 2: Develop Reports | Practice 2: Create Cards Regions and Faceted Search

| Name | Value |
|----------------|--------------|
| P17_PRODUCT_ID | &PRODUCT_ID. |

- Click **OK**.
- For Display Type, select **Text with Icon**.
- For Icon, select **fa-cart-edit**.

Click **Save**.

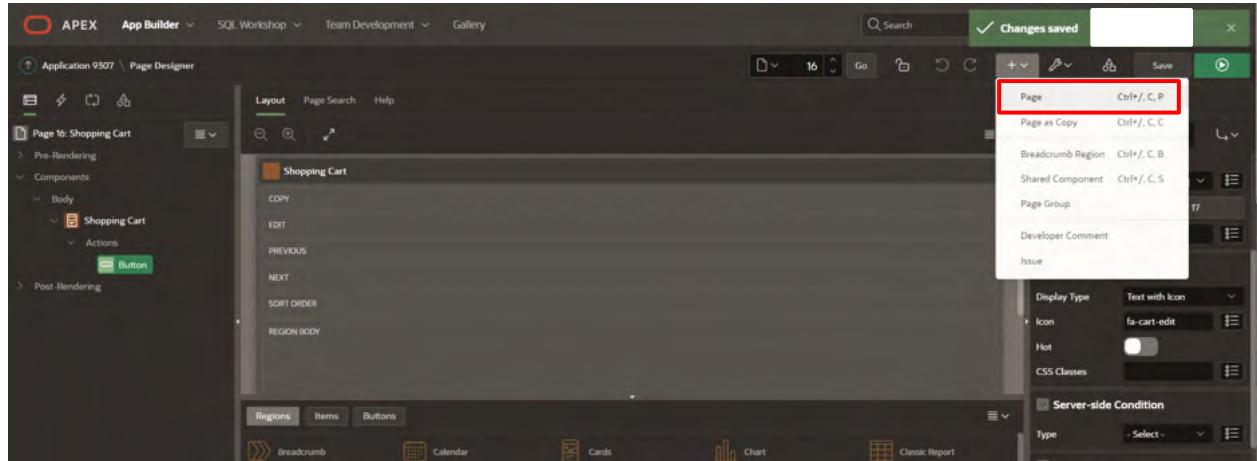


This configures the (Edit) button to open page 17, passing the value of the PRODUCT_ID column of the current card as the value for the P17_PRODUCT_ID page item in the called page.

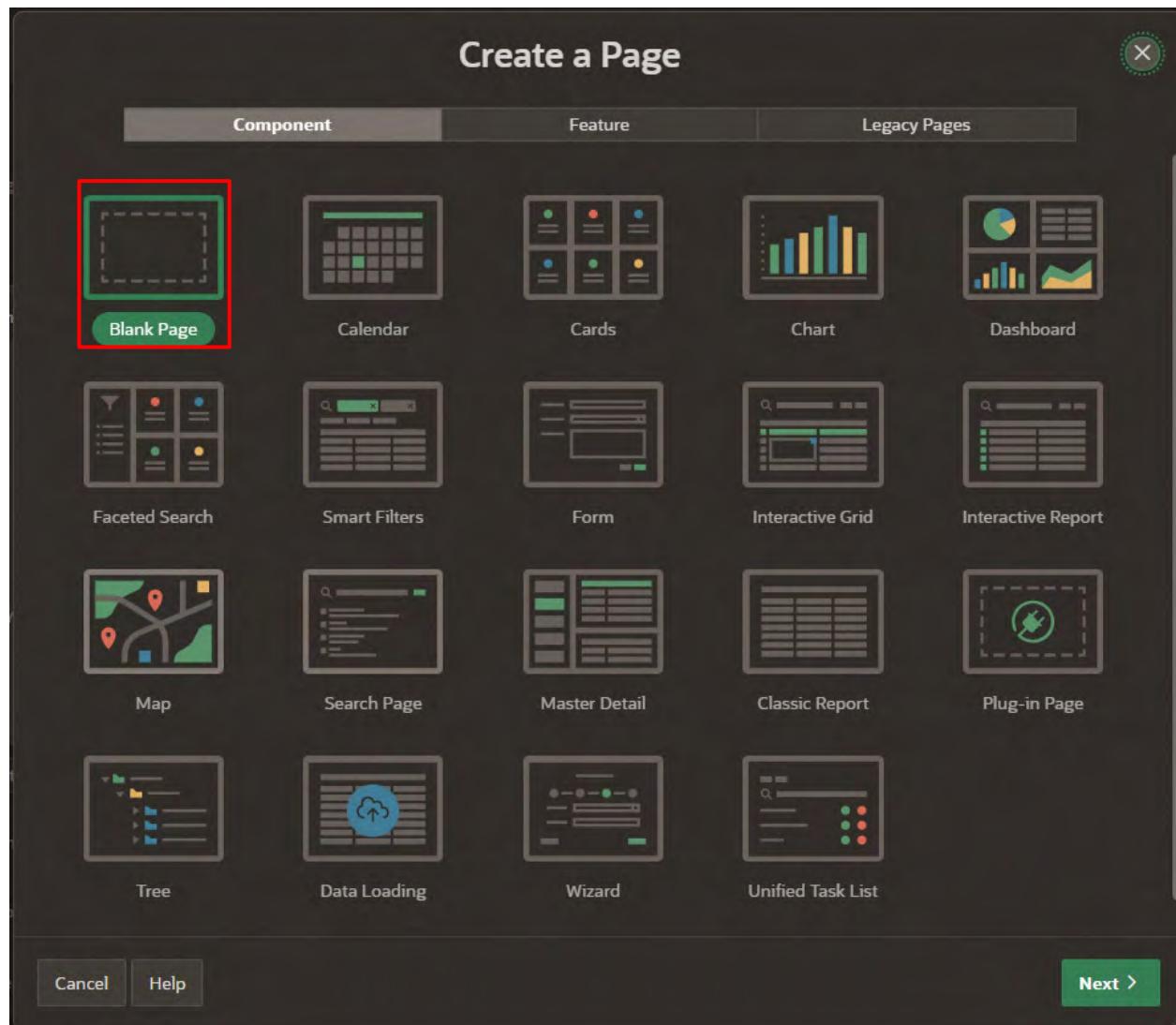
Create a Modal Page

This is to add products to the cart.

1. Navigate to the Create (+) button and click **Page**.



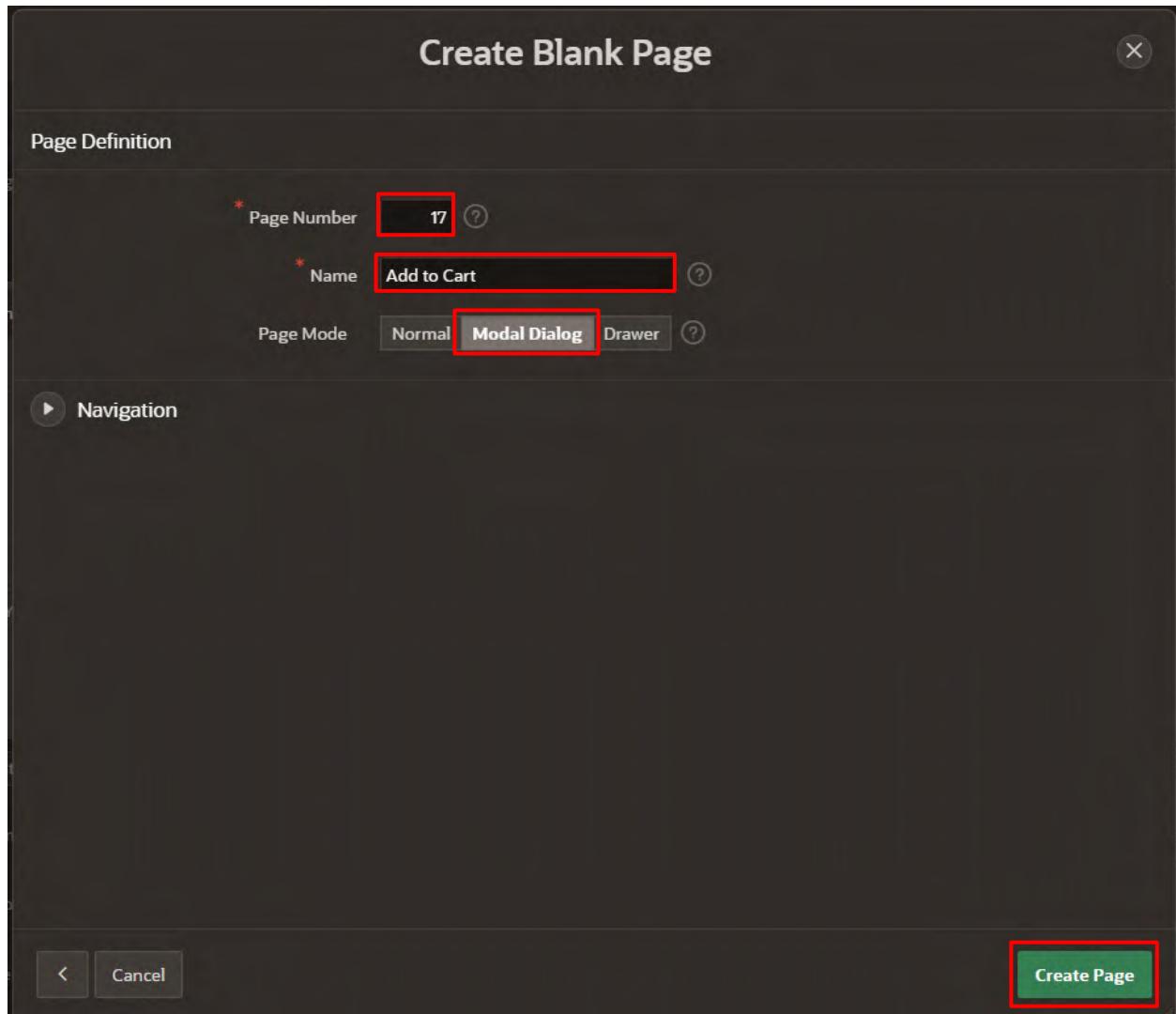
2. Select **Blank Page** and click **Next**.



3. Enter/select the following and click **Create Page**.

- Page Number: **17**
- Name: **Add to Cart**
- Page Mode: **Modal Dialog**

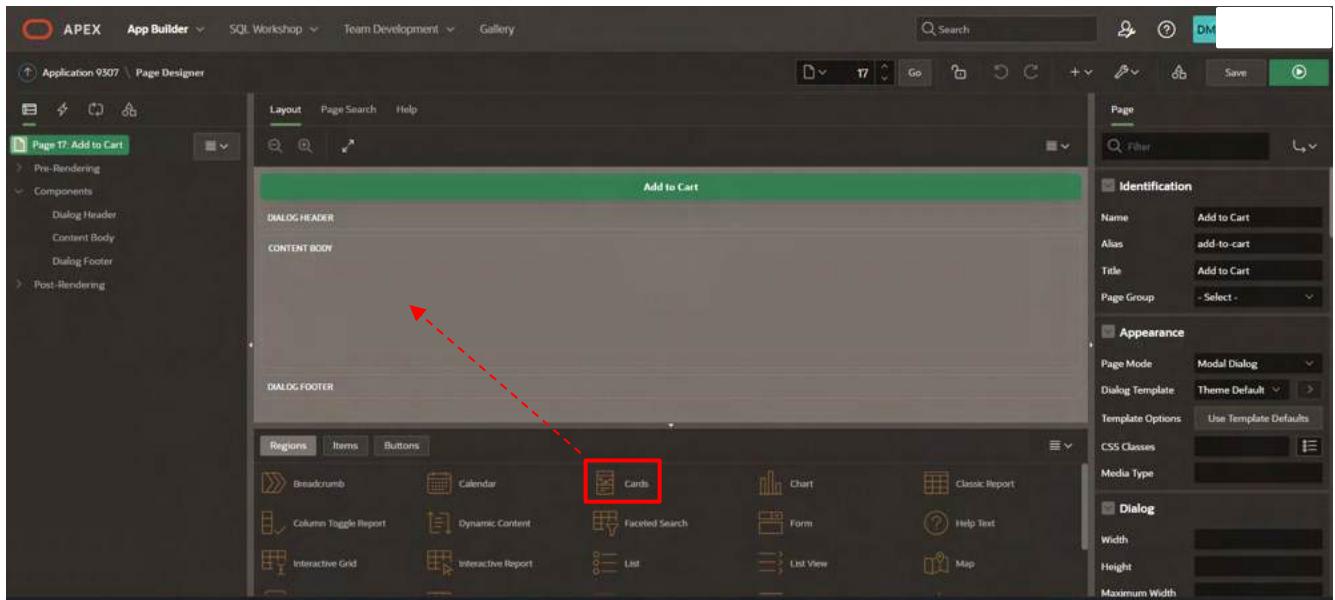
Click **Create Page**.



Add a Cards Region for Product Details

This region allows users to review the details of a product, such as brand, price, description, and more.

1. On the new modal page created, navigate to the **Gallery Menu**.
2. Drag a **Cards** region and drop it into the Content Body section.



3. In the Property Editor, enter/select the following:

- Title: **Product**

Under **Source**:

- Type: **SQL Query**
- SQL Query:

```

SELECT product_id,
       product_name,
       unit_price,
       product_details,
       product_image,
       image_mime_type,
       image_filename,
       image_charset,
       image_last_updated,
       color_id,
       department_id,
       clothing_id,
       d.description,
       b.brand
  FROM   products p,
         json_table (p.product_details, '$' columns (
description varchar2(4000) path '$.description') ) d,
         json_table (p.product_details, '$' columns ( brand
varchar2(4000) path '$.brand') ) b
 WHERE  product_id = :p17_product_id
    
```

The screenshot shows the 'Region' configuration dialog in Oracle APEX. The 'Identification' section has 'Title' set to 'Product'. The 'Source' section has 'Location' set to 'Local Database' and 'Type' set to 'SQL Query'. The 'SQL Query' field contains the following code:

```

SELECT product_id,
       product_name,
       unit_price,
       product_details,
       product_image,
       image_mime_type,
       image_filename,
       image_charset,
       image_last_updated,
       color_id,
       department_id,
       clothing_id,
       d.description,
       b.brand
  FROM products p,
       json_table (p.product_details,
       '$' columns ( description varchar2(4000)
                      path '$.description') ) d.

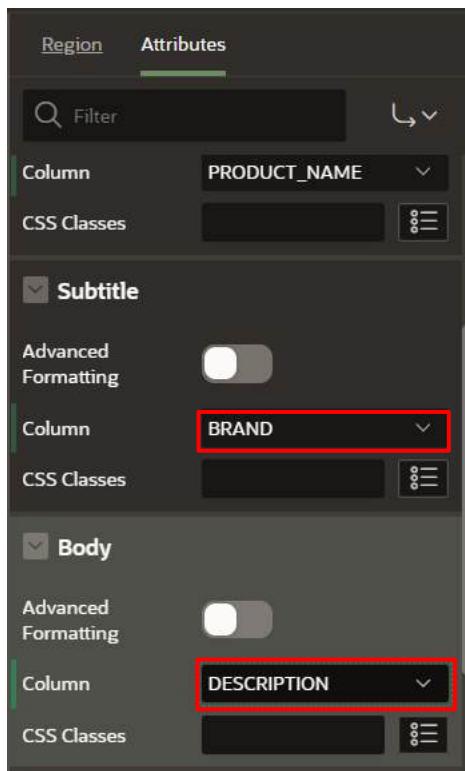
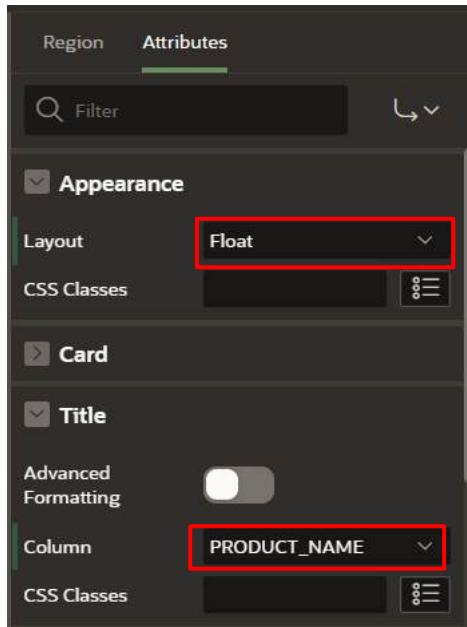
```

- Template Options: **Use Template Defaults**
- Style: **Style C**
- Click **OK**.

The screenshot shows the Oracle APEX Page Designer interface. On the left, the page structure is visible with components like 'Pre-Rendering', 'Components' (with 'Product' selected), and 'Post-Rendering'. In the center, a 'Template Options' dialog is open for the 'Product' component, specifically for the 'Style' tab, which is set to 'Style C'. On the right, the 'Region' configuration dialog is open, showing the 'Template Options' section with 'Use Template Defaults' checked. The status bar at the bottom indicates 'Copyright © 2023, Oracle and/or its affiliates.'

4. Click **Attributes** and select the following:

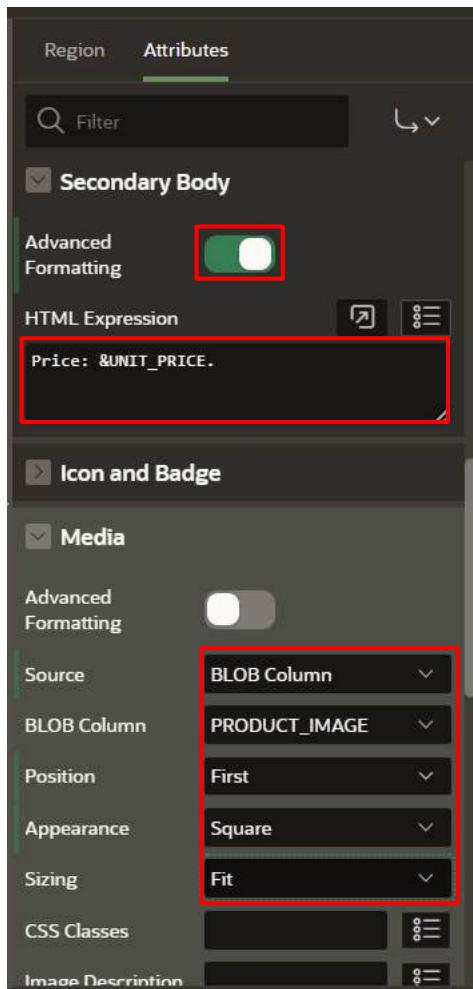
- Under **Appearance**: For Layout, select **Float**.
- Under **Title**: For Column, select **PRODUCT_NAME**.
- Under **Subtitle**: For Column, select **BRAND**.
- Under **Body**: For Column, select **DESCRIPTION**.



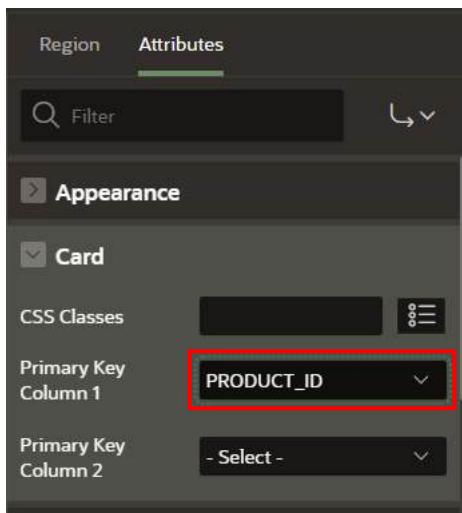
- Under **Secondary Body**:

- 1) Set Advanced Formatting to **On**. For HTML Expression, enter the following:

```
Price: &UNIT_PRICE.
```



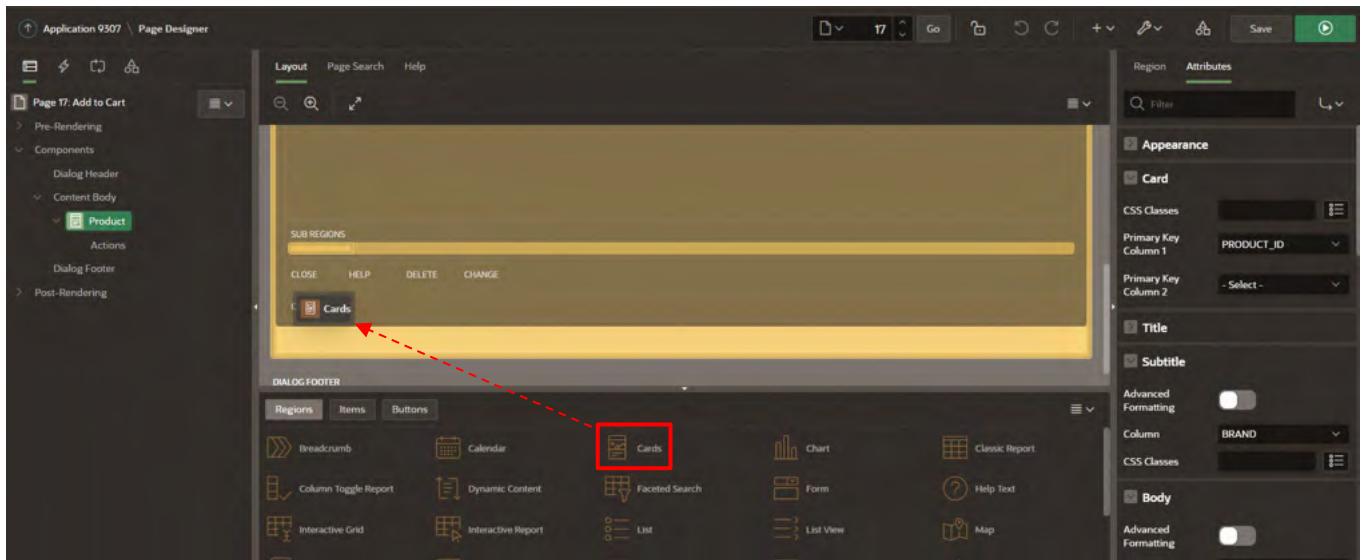
- Under **Card**: For Primary Key Column 1, select **PRODUCT_ID**.



Add a Cards Region for Customer Reviews

This region lets users read the customer reviews for a product.

1. Navigate to the **Gallery Menu**.
2. Drag a **Cards** region and drop it into the Content Body section under the **Product** region.

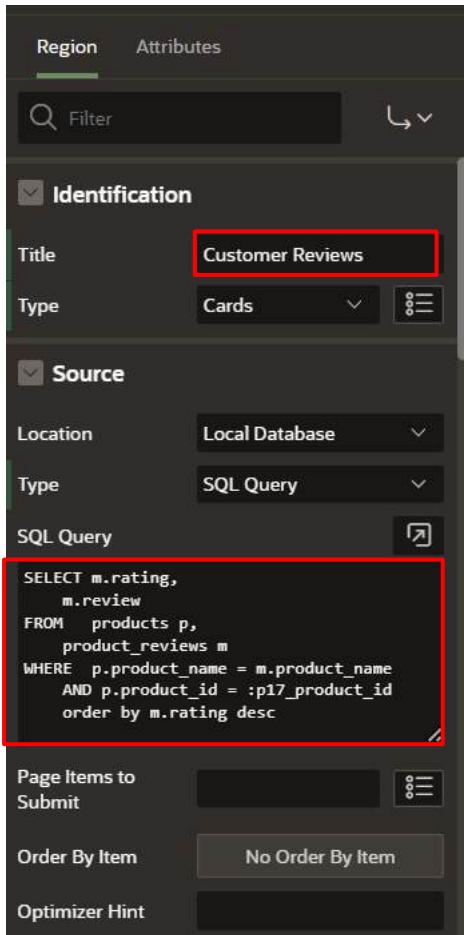


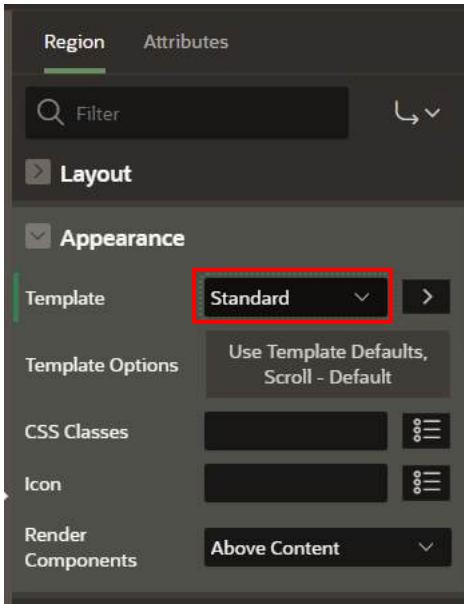
3. In the Property Editor, enter/select the following:
 - Title: **Customer Reviews**

Under Source:

- Type: **SQL Query**
- SQL Query:

```
SELECT m.rating,  
       m.review  
  FROM   products p,  
         product_reviews m  
 WHERE  p.product_name = m.product_name  
    AND p.product_id = :p17_product_id  
    ORDER BY m.rating DESC
```





4. Click **Attributes** and enter/select the following:

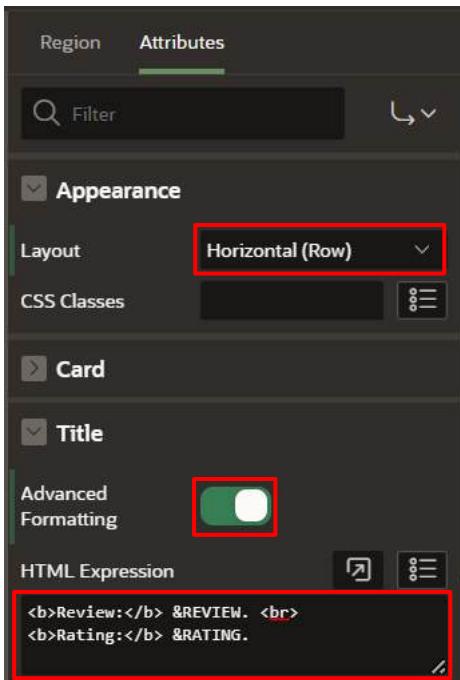
Under Appearance:

- Layout: **Horizontal (Row)**

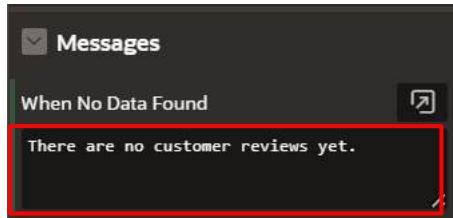
Under Title:

- Advanced Formatting: **On**
- HTML Expression:

```
<b>Review:</b> &REVIEW. <br>
<b>Rating:</b> &RATING.
```



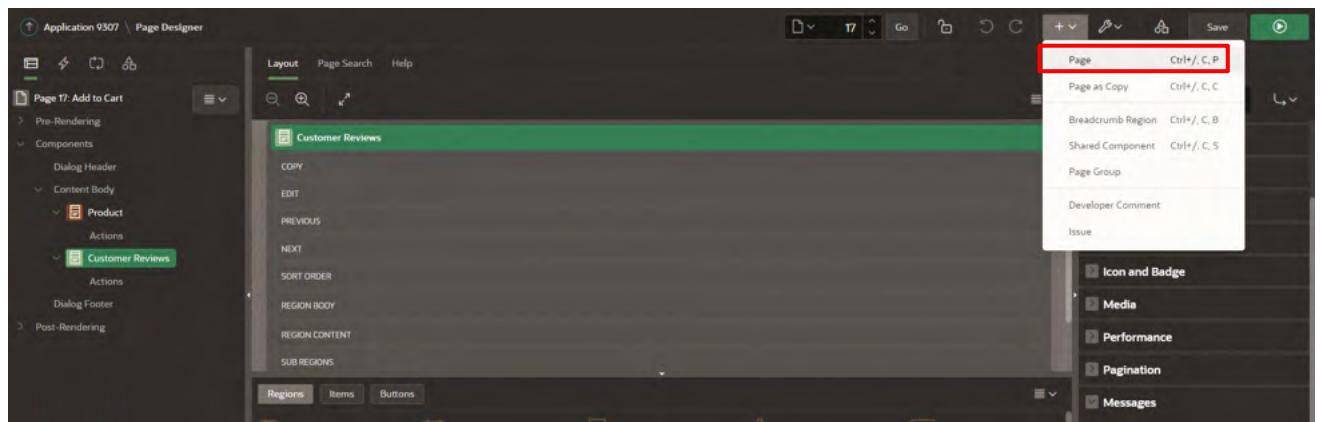
- Under Messages:
- When No Data Found: **There are no customer reviews yet.**



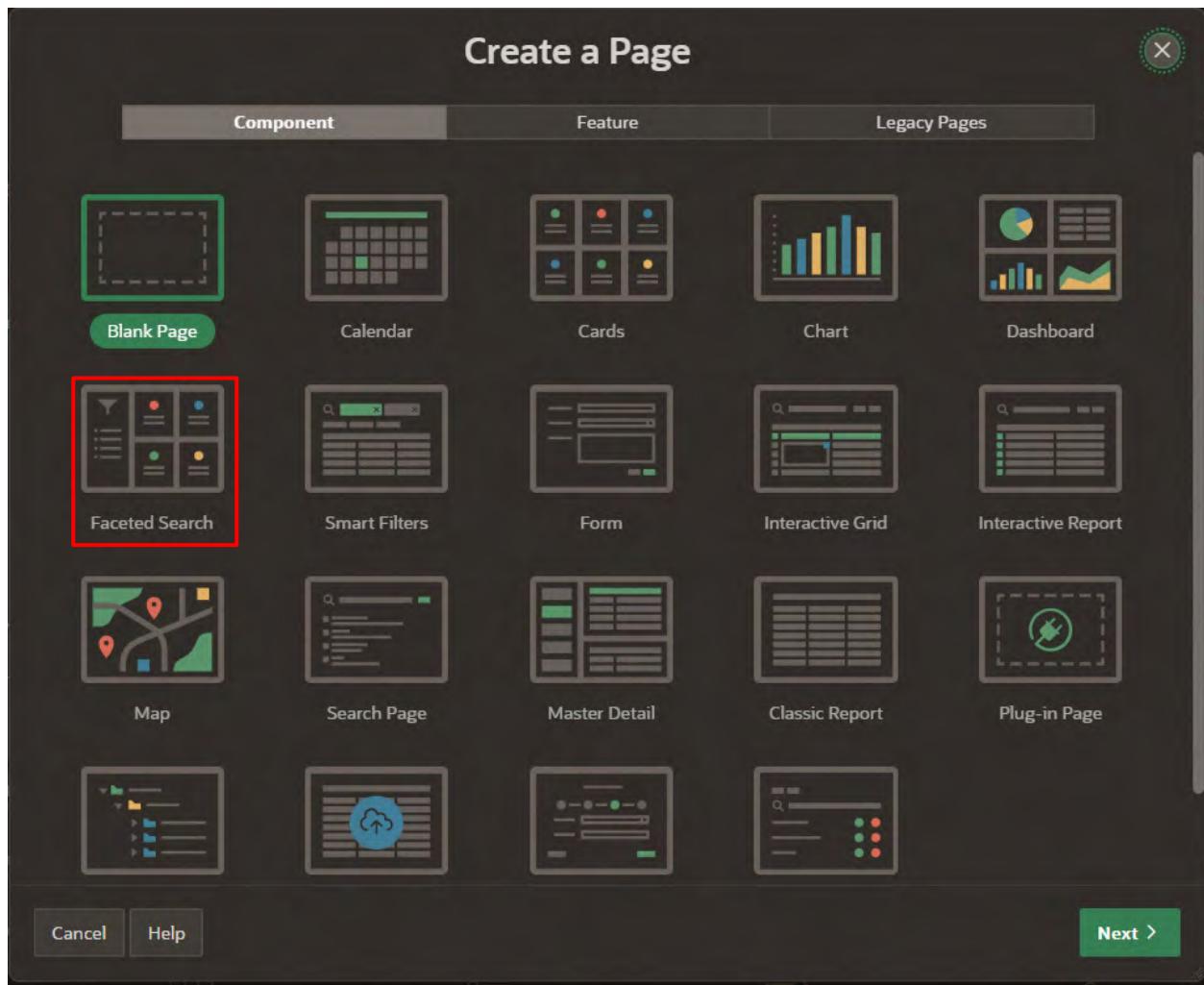
5. Click **Save**.

Add the Products Page

1. Navigate to the Create button (+) and click **Page**.



2. Select **Faceted Search** and then click **Next**.



3. Enter/select the following and click **Next**:

Under Page Definition:

- Page Number: **19**
- Name: **Products**

Under Data Source:

- Table/View Name: **PRODUCTS**

Under Navigation:

- Use Breadcrumb: **No**

Create Faceted Search

Page Definition

* Page Number [?](#)

* Name [?](#)

Data Source

Data Source Local Database REST Enabled SQL Service REST Data Source [?](#)

Source Type Table SQL Query [?](#)

* Table / View Owner [?](#)

* Table / View Name [?](#)

Navigation

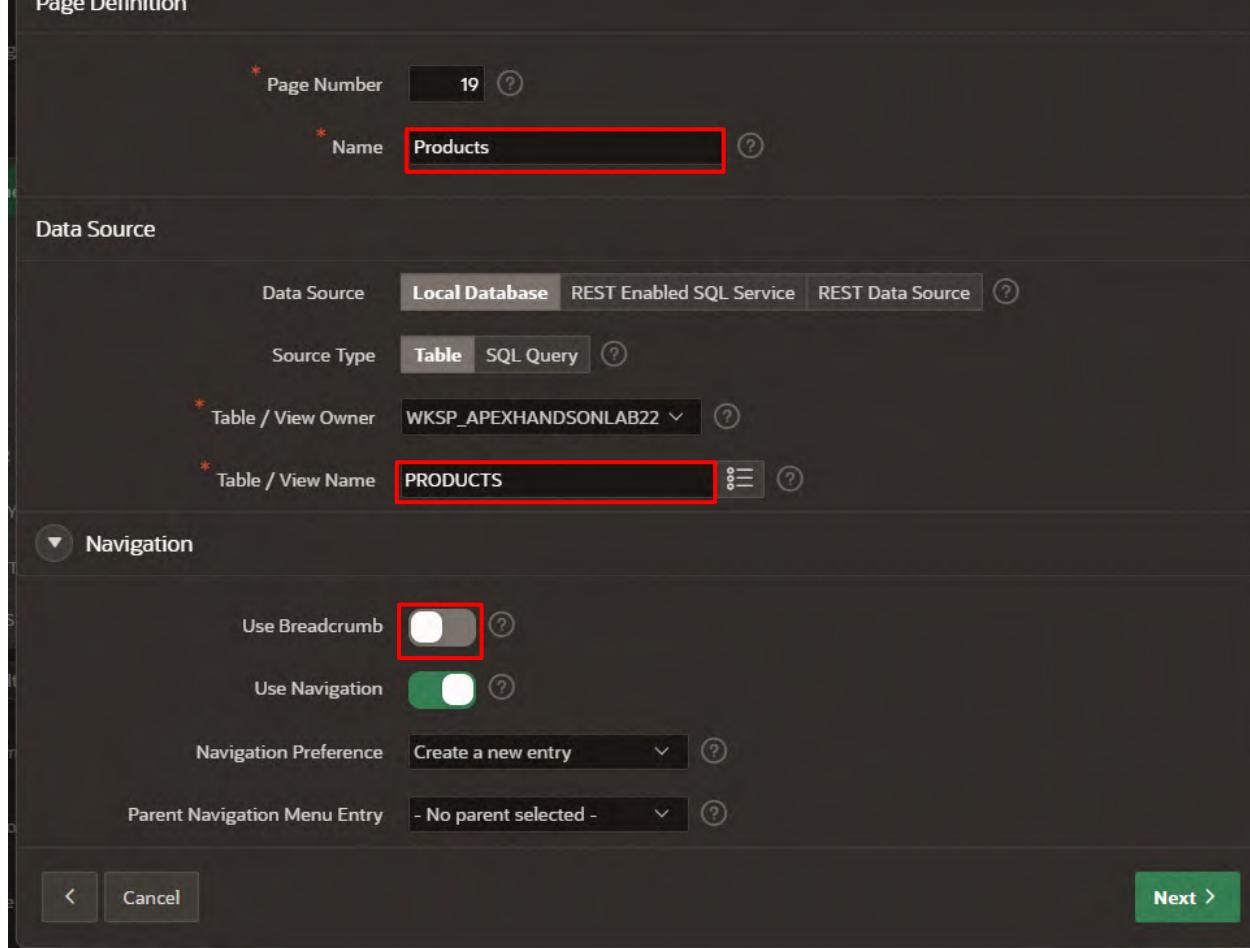
Use Breadcrumb [?](#)

Use Navigation [?](#)

Navigation Preference [?](#)

Parent Navigation Menu Entry [?](#)

[Cancel](#) [Next >](#)



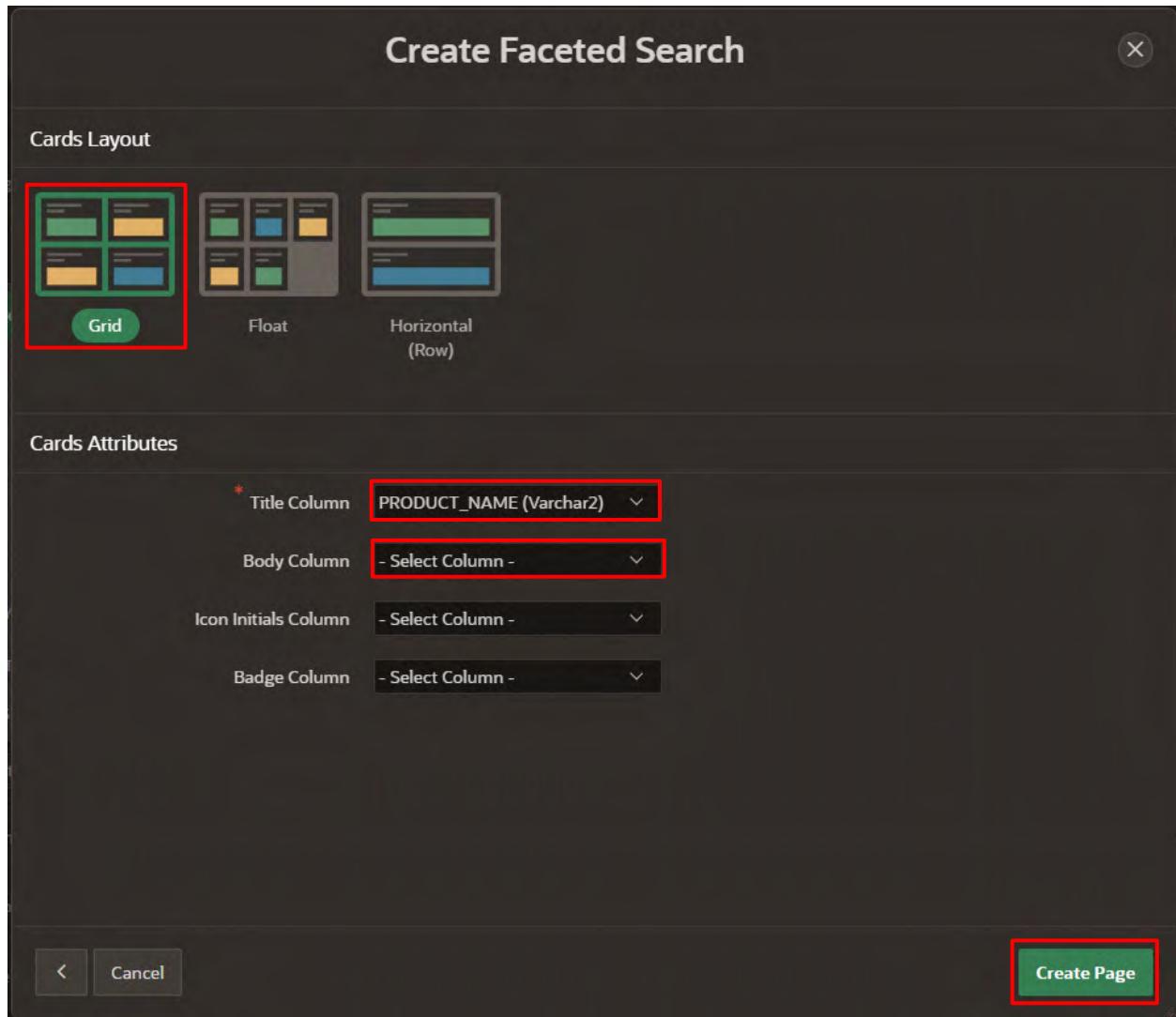
Click **Next**.

4. For Facet Selection, next to **Display as**, select **Cards**. Leave the remaining options as their defaults and click **Next**.

| Column | Facet | Searchable | Nulls | Distinct | Average Length |
|--|----------|------------|-------|----------|----------------|
| PRODUCT_ID (number) | Range | No | 0 | 46 | 3 |
| PRODUCT_NAME (varchar2) | Range | Yes | 0 | 46 | 21 |
| <input checked="" type="checkbox"/> UNIT_PRICE (number) | Range | No | 0 | 46 | 4 |
| PRODUCT_DETAILS (blob) | Range | No | 0 | 0 | 794 |
| PRODUCT_IMAGE (blob) | Range | No | 0 | 0 | 134 |
| IMAGE_MIME_TYPE (varchar2) | Range | Yes | 46 | 0 | 0 |
| IMAGE_FILENAME (varchar2) | Range | Yes | 46 | 0 | 0 |
| IMAGE_CHARSET (varchar2) | Range | Yes | 46 | 0 | 0 |
| IMAGE_LAST_UPDATED (date) | Range | No | 46 | 0 | 1 |
| <input checked="" type="checkbox"/> COLOR_ID (number) | Checkbox | Yes | 0 | 7 | 3 |
| <input checked="" type="checkbox"/> DEPARTMENT_ID (number) | Checkbox | Yes | 0 | 4 | 3 |
| <input checked="" type="checkbox"/> CLOTHING_ID (number) | Checkbox | Yes | 0 | 12 | 3 |

5. Under **Create Faceted Search**, select the following:

- Select **Grid**
- Title Column: **PRODUCT_NAME**
- Body Column: - **Select Column** - (to deselect the default column chosen)
- Click Create Page.



You now know how to create an application item, an application process, a Faceted Search page, and a Cards page. You may now **proceed to the next practice**.

Practice: Managing and Customizing Interactive Reports

Practice 1: Customize an Interactive Report as a Developer

Overview

In this practice, you will manage and customize the Interactive Report page in the **Sample Reports** application as a **Developer**.

Downloads

- Did you miss out on trying the previous practices? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, follow the steps described in the [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Tasks

Customize an Interactive Report as a Developer

- In this practice, you edit an interactive report in page designer and customize it for end users. First, view the Interactive Report in page designer. In the Developer Toolbar, click **Edit Page**.

The screenshot shows the 'Interactive Report' page from the 'Sample Reporting' application. The page title is 'Primary Report'. The main content is a table listing various tasks across different projects. The columns include Project, Task Name, Start Date, End Date, Status, Assigned To, Cost, Budget, and Available Budget. The table has 15 rows. At the bottom of the page, there is a toolbar with several icons: a magnifying glass, a refresh symbol, a list icon, a search icon, a 'Session' button, a 'Debug' button, a 'Quick Edit' button, a 'Customize' button, and a help icon. The 'Page 1' button is highlighted with a red box.

| Project | Task Name | Start Date | End Date | Status | Assigned To | Cost | Budget | Available Budget |
|------------------------------|---|------------|-----------|---------|---------------|-------|--------|------------------|
| Bug Tracker | Document quality assurance procedures | 20-DEC-22 | 23-DEC-22 | Closed | Myra Sutcliff | 3,000 | 2,000 | -1,000 |
| Bug Tracker | Review automated testing tools | 24-DEC-22 | 26-DEC-22 | Closed | Myra Sutcliff | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | 08-JAN-23 | 08-JAN-23 | Closed | Myra Sutcliff | 100 | 100 | 0 |
| Discussion Forum | Identify owners | 09-JAN-23 | 09-JAN-23 | Closed | Hank Davis | 250 | 300 | 50 |
| Maintain Support Systems | HR software upgrades | 12-JAN-23 | 15-JAN-23 | Closed | Pam King | 8,000 | 7,000 | -1,000 |
| Discussion Forum | Install ACME Web Express application on internet server | 15-JAN-23 | 15-JAN-23 | Closed | Hank Davis | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | 15-JAN-23 | 20-JAN-23 | On-Hold | Myra Sutcliff | 1,000 | 2,000 | 1,000 |
| Maintain Support Systems | Apply Billing System updates | 16-JAN-23 | 18-JAN-23 | Closed | Russ Saunders | 9,500 | 7,000 | -2,500 |
| Public Website | Determine host server | 19-JAN-23 | 19-JAN-23 | Closed | Tiger Scott | 200 | 200 | 0 |
| Public Website | Check software licenses | 19-JAN-23 | 19-JAN-23 | Closed | Tom Suess | 100 | 100 | 0 |
| Employee Satisfaction Survey | Complete questionnaire | 19-JAN-23 | 20-JAN-23 | Closed | Irene Jones | 1,200 | 800 | -400 |
| Maintain Support Systems | Arrange for vacation coverage | 20-JAN-23 | 20-JAN-23 | Open | Al Bines | 300 | 500 | 200 |
| Public Website | Purchase additional software | | | | | 300 | 1,000 | 700 |

2. When end users click the edit icon for a specified row, they should be directed to a page that shows the column values for that row. The interactive report currently has a link column to a form page. Modify your interactive report to have a link to a single row view.
- Under Rendering, navigate to the **Projects** interactive report and then on the Property Editor, select **Attributes**. Under **Link**, for **Link Column**, select **Link to Single Row View**. Click **Save** and **Run Page**.

The screenshot shows the Oracle Application Express Page Designer interface. On the left, the navigation tree is open, showing 'Page 1: Interactive Report' under 'Components' with 'Projects' selected. The main area displays the 'Projects' report with a green header bar. On the right, the 'Properties' panel is open, specifically the 'Link' section. A red box highlights the 'Link Column' dropdown, which is set to 'Link to Single Row View'. Other settings in the 'Link' section include 'Target' (Link to Custom Target) and 'Link Icon' (an edit pencil icon). Below the 'Link' section, other properties like 'Authorization Scheme' and 'Condition Type' are visible.

- In the report, ensure that you have selected **1. Primary Report** in the report toolbar. Click the **edit icon (pencil)** for any row.

The screenshot shows the 'Interactive Report' page. At the top, there is a search bar and a toolbar with various icons. Below the toolbar is a table with columns: Project, Task Name, Start Date, End Date, Status, Assigned To, Cost, Budget, and Available Budget. The first row of the table is highlighted with a red box around its edit icon. The table contains 15 rows of task data, such as 'Bug Tracker' and 'Discussion Forum' tasks with their respective details and assigned team members.

- c. The details of maintaining the project are displayed. Click the **Cancel** button to return to the report.

Maintain Project

Project: Maintain Support Systems

Task Name: HR software upgrades

Start Date: 12-JAN-23

End Date: 15-JAN-23

Status: Closed

Assigned To: Pam King

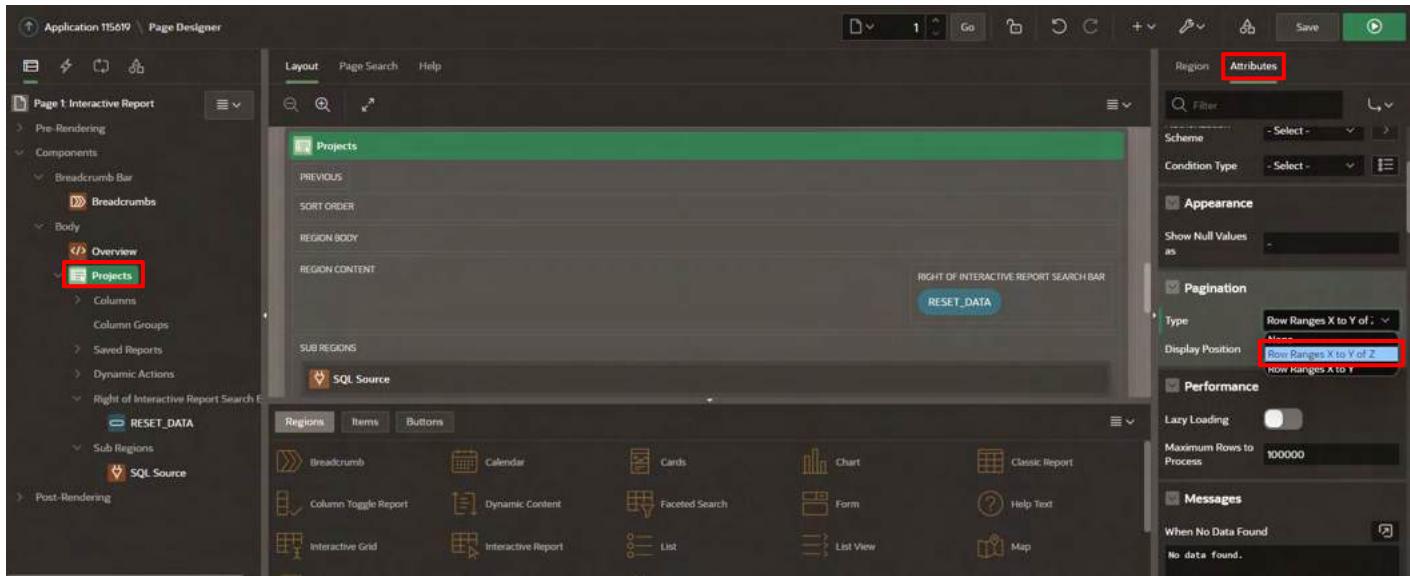
Cost: 8000

Budget: 7000

Cancel Delete Apply Changes

3. The current pagination type of the interactive report is **Row Ranges X to Y**. Assume you want to always show your end users the total number of rows in the report as part of the pagination display. In the Developer Toolbar, click **Edit Page**.
- a. Under Rendering, navigate to the **Projects** interactive report and select **Attributes** in the **Property Editor**.

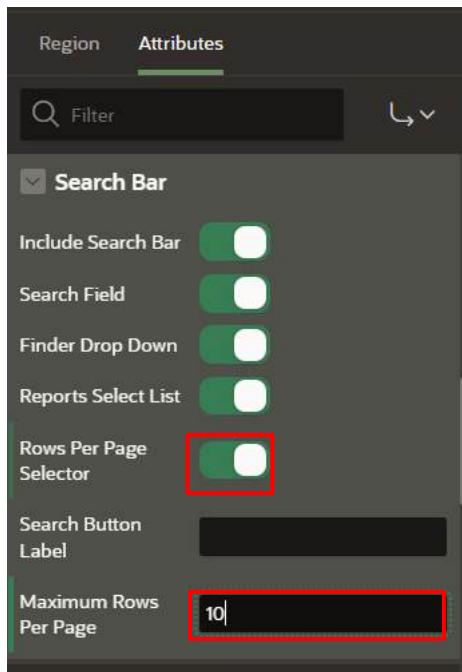
- b. In the Property Editor, under **Attributes**, search for **Pagination**. For Type, **select Row Ranges X to Y of Z**.



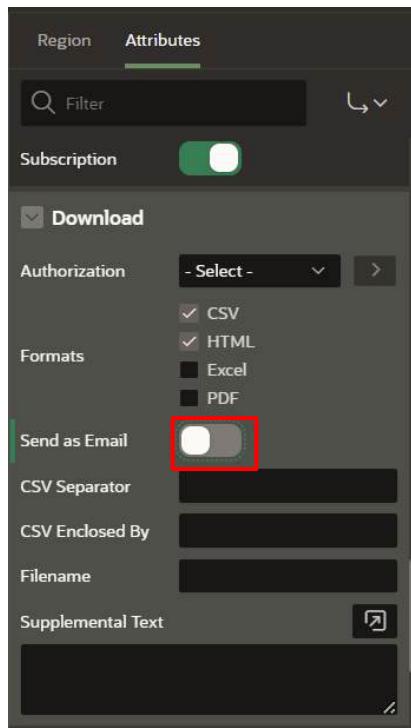
4. You want to customize the display of the Search Bar. End users should be able to select the display of the desired number of rows per page.

In the Property Editor, locate **Search Bar** under **Attributes**. Perform the following:

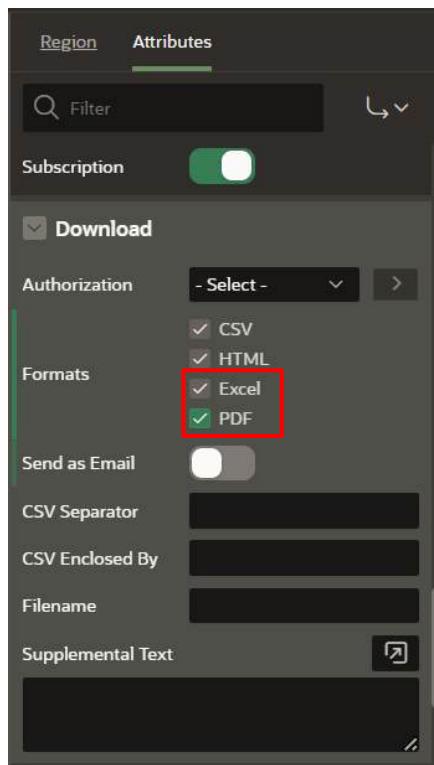
- Enable Rows Per Page Selector.
- Enter **10** for Maximum Rows Per Page.



5. You want to disable Email in the Download option of the Actions menu. In the Property Editor, navigate to **Download**. Deselect the **Send as Email** download format.



6. You want to enable **Excel** and **PDF** formats in the Download option of the Actions menu. In the Property Editor, navigate to **Download**. Select the **Excel** and **PDF** download formats.



- Now that you have finished the customization for end users, click **Save** and **Run Page**.
- Notice that the **row selector** and the **new pagination type** are available in the report.

The screenshot shows an 'Interactive Report' interface. At the top, there's a search bar, a 'Go' button, and a dropdown for 'Primary Report'. Below that is a toolbar with icons for search, refresh, and actions. A dropdown menu labeled 'Rows' is open, showing options 1, 5, and 10, with '10' selected. To the right of the table is a 'Actions' dropdown menu with various options like Filter, Data, Format, Chart, Group By, Pivot, Report, Save Report, Download, Reset, Subscription, Help, and a 'Customize' icon. The main area displays a table of tasks with columns for Project, Task Name, Start Date, End Date, Status, Assigned To, Cost, Budget, and Available Budget. The table has 10 rows of data. At the bottom, there's a footer with links for Home, App 115619, Page 1, Session, Debug, Quick Edit, Customize, and Help.

- Select **Actions > Report > Save Report**.

The screenshot shows the same 'Interactive Report' interface as before, but with a different focus. The 'Report' option in the dropdown menu is highlighted with a red box. The rest of the menu items are visible but not highlighted. The table of tasks is still present below the menu.

- The **Save Report** dialog box is displayed. Notice that the **Public** check box is now available. Perform the following:
 - Save: Select **As Named Report**
 - Name: Enter **Projects Public Report**
 - Enable the **Public** check box.

Save Report

Save (Only displayed for developers)

As Named Report

Name

Projects Public Report

Description

Public

Cancel Apply

- This report is now saved as a public report for all users and is available in the Reports drop-down list.

Interactive Report

| Project | Task | Start Date | End Date | Status | Assigned To | Cost | Budget | Available Budget |
|--------------------------|---|------------|-----------|---------|---------------|-------|--------|------------------|
| Bug Tracker | Public | 20-DEC-22 | 23-DEC-22 | Closed | Myra Sutcliff | 3,000 | 2,000 | -1,000 |
| Bug Tracker | Review automated testing tools | 24-DEC-22 | 26-DEC-22 | Closed | Myra Sutcliff | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | 08-JAN-23 | 08-JAN-23 | Closed | Myra Sutcliff | 100 | 100 | 0 |
| Discussion Forum | Identify owners | 09-JAN-23 | 09-JAN-23 | Closed | Hank Davis | 250 | 300 | 50 |
| Maintain Support Systems | HR software upgrades | 12-JAN-23 | 15-JAN-23 | Closed | Pam King | 8,000 | 7,000 | -1,000 |
| Discussion Forum | Install ACME Web Express application on internet server | 15-JAN-23 | 15-JAN-23 | Closed | Hank Davis | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | 15-JAN-23 | 20-JAN-23 | On-Hold | Myra Sutcliff | 1,000 | 2,000 | 1,000 |
| Maintain Support Systems | Apply Billing System updates | 16-JAN-23 | 18-JAN-23 | Closed | Russ Saunders | 9,500 | 7,000 | -2,500 |
| Public Website | Determine host server | 19-JAN-23 | 19-JAN-23 | Closed | Tiger Scott | 200 | 200 | 0 |
| Public Website | Check software licenses | 19-JAN-23 | 19-JAN-23 | Closed | Tom Suess | 100 | 100 | 0 |

1 - 10 of 73

SQL Source Home App 115619 Page 1 Session Debug Quick Edit Customize Close

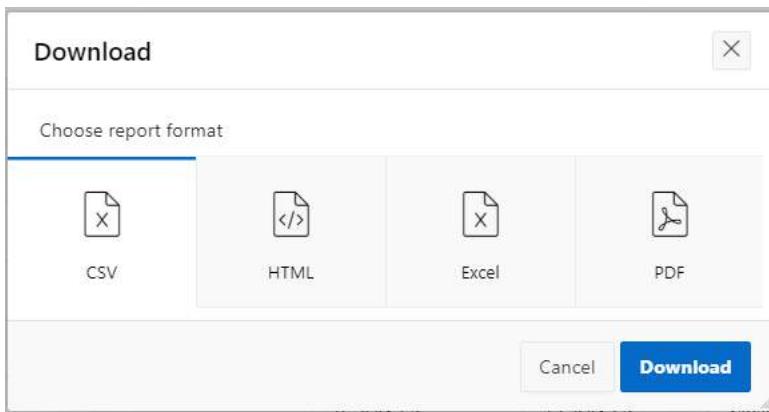
- Click **Actions > Download**.

- Notice that the **Excel** and **PDF** formats are available. Click the **Close** icon.

Interactive Report

The screenshot shows a database report titled "1. Projects Public Report". The report lists tasks categorized by project. A context menu is open over the last task in the list, showing options like Filter, Data, Format, Chart, Group By, Pivot, Report, Download (which is highlighted with a red box), Subscription, and Help.

| Project | Task Name | End Date | Status | Assigned To | Cost | Budget | Available Budget |
|--------------------------|---|-----------|---------|---------------|-------|--------|------------------|
| Bug Tracker | Document quality assurance procedures | 23-DEC-22 | Closed | Myra Sutcliff | 3,000 | 2,000 | -1,000 |
| Bug Tracker | Review automated testing tools | 26-DEC-22 | Closed | Myra Sutcliff | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | 08-JAN-23 | Closed | Myra Sutcliff | 100 | 100 | 0 |
| Discussion Forum | Identify owners | 09-JAN-23 | Closed | Hank Davis | 250 | 300 | 50 |
| Maintain Support Systems | HR software upgrades | 15-JAN-23 | Closed | Pam King | 8,000 | 7,000 | -1,000 |
| Discussion Forum | Install ACME Web Express application on internet server | 15-JAN-23 | Closed | Hank Davis | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | 20-JAN-23 | On-Hold | Myra Sutcliff | 1,000 | 2,000 | 1,000 |
| Maintain Support Systems | Apply Billing System updates | 18-JAN-23 | Closed | Russ Saunders | 9,500 | 7,000 | -2,500 |
| Public Website | Determine host server | 19-JAN-23 | Closed | Tiger Scott | 200 | 200 | 0 |
| Public Website | Check software licenses | 19-JAN-23 | Closed | Tom Suess | 100 | 100 | 0 |



14. From the Reports drop-down list, select **Primary Report**.

You now know how to manage and customize the interactive report as a developer. You may now **proceed to the next practice**.

Practice 2: Customize an Interactive Report as an End User

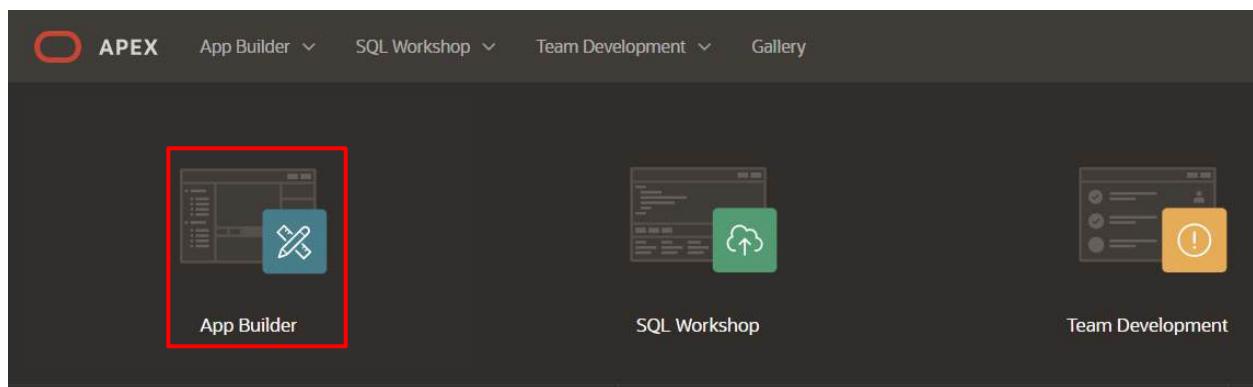
Overview

In this practice, you will manage and customize the Interactive Report page in the **Sample Reports** application as an **End User**.

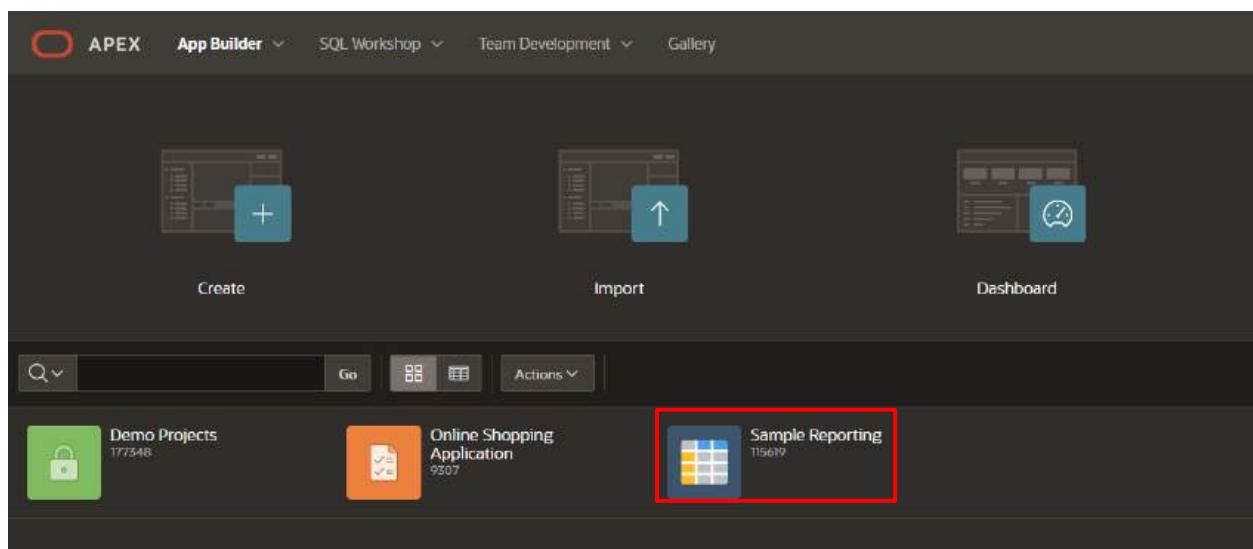
Tasks

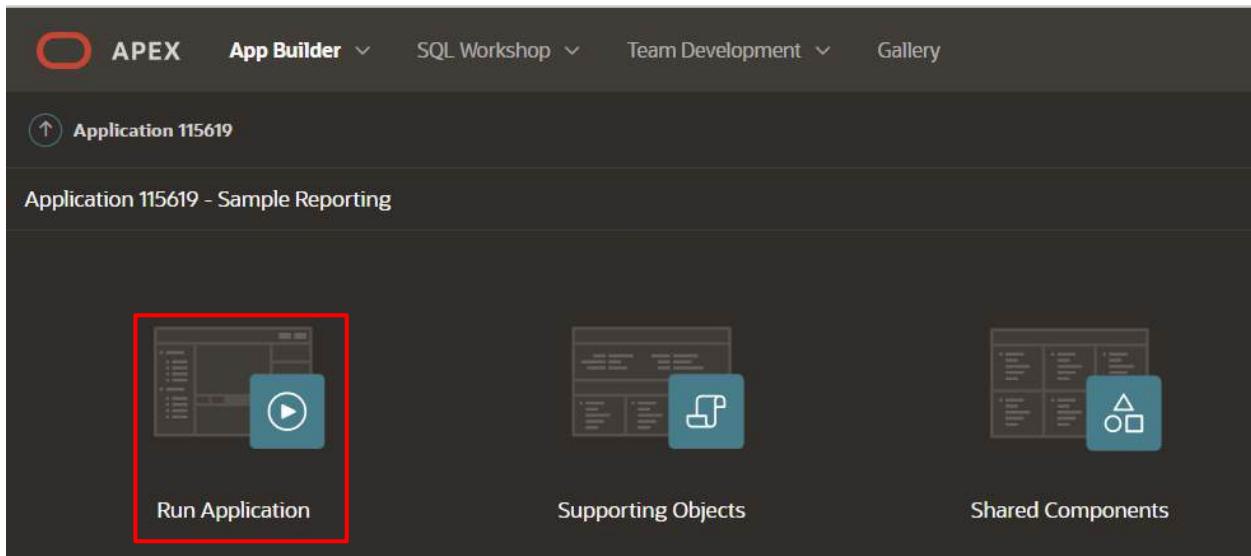
Customize an Interactive Report as an End User

1. To start, run the **Sample Reporting** application. Click **App Builder**.



2. Navigate to **Sample Reporting** and then click **Run Application**.





- In the **navigation menu** or in the **Cards Region**, click **Interactive Report**.

The screenshot shows the 'Sample Reporting' application page. On the left, there's a sidebar with a navigation menu. The 'Interactive Report' item is highlighted with a red box. The main content area has a title 'Sample Reporting' with a subtitle 'Demonstration of reports and reporting techniques in Oracle APEX'. Below that is an 'About' section with a brief description. To the right, there are several cards. One card, 'Interactive Report', is highlighted with a red box. Other cards include 'Interactive Grid', 'Faceted Search', 'Cards', 'Classic Report', 'Use Cases', 'SQL Examples', 'Analytic Functions', and 'Administration'.

- Assume that you do not want to display the **Start Date** and **End Date** columns in the report. Also, you want the **Status** column to display just after the **Assigned To** column. Perform the following steps:

- Click **Actions** and select **Columns**.

Sample Reporting

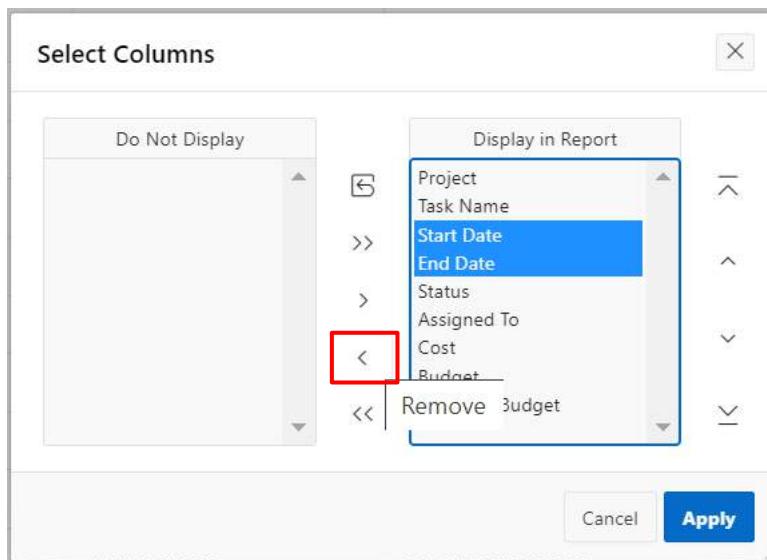
Interactive Report

icon. You can sort and filter columns by clicking on column headings, click the Actions drop down menu to control the columns to display on the report and the order in which they are displayed. You can also save custom reports, chart data, and perform many other actions.

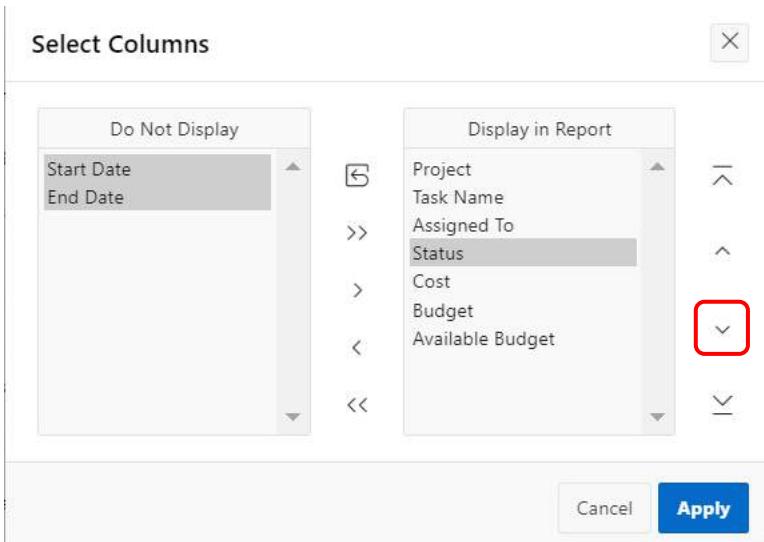
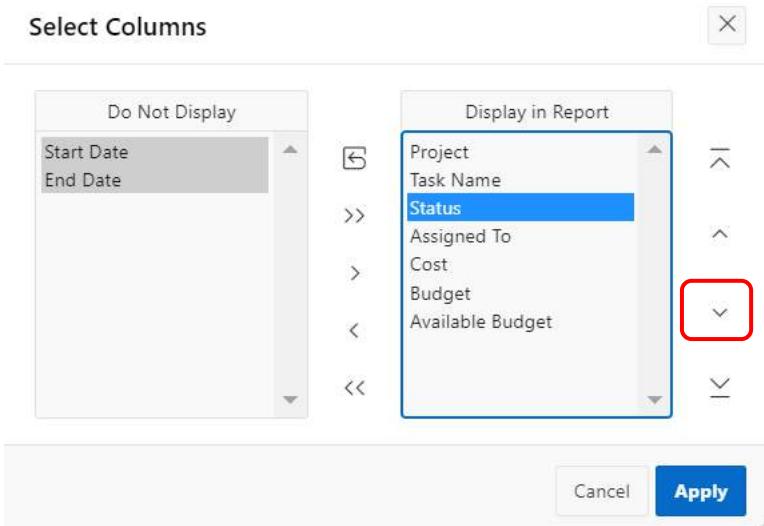
| Project | Task Name | Start Date | Columns | Status | Assigned To | Cost | Budget | Available Budget |
|--------------------------|---|------------|--------------|--------|---------------|-------|--------|------------------|
| Bug Tracker | Document quality assurance procedures | 20-DEC-22 | Filter | osed | Myra Sutcliff | 3,000 | 2,000 | -1,000 |
| Bug Tracker | Review automated testing tools | 24-DEC-22 | Data | osed | Myra Sutcliff | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | 08-JAN-23 | Format | osed | Myra Sutcliff | 100 | 100 | 0 |
| Discussion Forum | Identify owners | 09-JAN-23 | Chart | osed | Hank Davis | 250 | 300 | 50 |
| Maintain Support Systems | HR software upgrades | 12-JAN-23 | Group By | osed | Pam King | 8,000 | 7,000 | -1,000 |
| Discussion Forum | Install ACME Web Express application on internet server | 15-JAN-23 | Pivot | osed | Hank Davis | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | 15-JAN-23 | Report | n-Hold | Myra Sutcliff | 1,000 | 2,000 | 1,000 |
| Maintain Support Systems | Apply Billing System updates | 16-JAN-23 | Download | osed | Russ Saunders | 9,500 | 7,000 | -2,500 |
| Public Website | Determine host server | 19-JAN-23 | Subscription | osed | Tiger Scott | 200 | 200 | 0 |
| Public Website | Check software licenses | 19-JAN-23 | Help | osed | Tom Suess | 100 | 100 | 0 |

1 < 10 of 73 >

- b. The **Select Columns** dialog box appears. The columns on the right are displayed, and the columns on the left are hidden. Select **Start Date** and **End Date** in the **Display in Report** group, and move the selected columns into the **Do Not Display** list by clicking the left arrow (<).



- c. You can reorder the displayed columns by using the arrows on the far right. Select **Status** in the Display in Report group and click the down arrow once. Then, click **Apply**.



- d. The interactive report is displayed with the changes made in the previous steps.

Interactive Report

| Project | | Task Name | Assigned To | Status | Cost | Budget | Available Budget |
|---------|--------------------------|---|---------------|---------|-------|--------|------------------|
| | Bug Tracker | Document quality assurance procedures | Myra Sutcliff | Closed | 3,000 | 2,000 | -1,000 |
| | Bug Tracker | Review automated testing tools | Myra Sutcliff | Closed | 750 | 1,500 | 750 |
| | Bug Tracker | Implement bug tracking software | Myra Sutcliff | Closed | 100 | 100 | 0 |
| | Discussion Forum | Identify owners | Hank Davis | Closed | 250 | 300 | 50 |
| | Maintain Support Systems | HR software upgrades | Pam King | Closed | 8,000 | 7,000 | -1,000 |
| | Discussion Forum | Install ACME Web Express application on internet server | Hank Davis | Closed | 100 | 100 | 0 |
| | Bug Tracker | Train developers on tracking bugs | Myra Sutcliff | On-Hold | 1,000 | 2,000 | 1,000 |
| | Maintain Support Systems | Apply Billing System updates | Russ Saunders | Closed | 9,500 | 7,000 | -2,500 |
| | Public Website | Determine host server | Tiger Scott | Closed | 200 | 200 | 0 |
| | Public Website | Check software licenses | Tom Suess | Closed | 100 | 100 | 0 |

1 - 10 of 73 >

SQL Source

5. Next, let's say you want to filter the report results. First, you want to filter the report to display rows that meet the criteria **Cost <= 2000**. Within these filtered results, you then create another filter to display rows with **Project is Bug Tracker**. Perform the following steps:

- a. Click **Actions** and select **Filter**.

Interactive Report

| Project | | Task Name | Status | Cost | Budget | Available Budget |
|---------|--------------------------|---|---------|-------|--------|------------------|
| | Bug Tracker | Document quality assurance procedures | Closed | 3,000 | 2,000 | -1,000 |
| | Bug Tracker | Review automated testing tools | Closed | 750 | 1,500 | 750 |
| | Bug Tracker | Implement bug tracking software | Closed | 100 | 100 | 0 |
| | Discussion Forum | Identify owners | Closed | 250 | 300 | 50 |
| | Maintain Support Systems | HR software upgrades | Closed | 8,000 | 7,000 | -1,000 |
| | Discussion Forum | Install ACME Web Express application on internet server | Closed | 100 | 100 | 0 |
| | Bug Tracker | Train developers on tracking bugs | On-Hold | 1,000 | 2,000 | 1,000 |
| | Maintain Support Systems | Apply Billing System updates | Closed | 9,500 | 7,000 | -2,500 |
| | Public Website | Determine host server | Closed | 200 | 200 | 0 |
| | Public Website | Check software licenses | Closed | 100 | 100 | 0 |

1 - 10 of 73 >

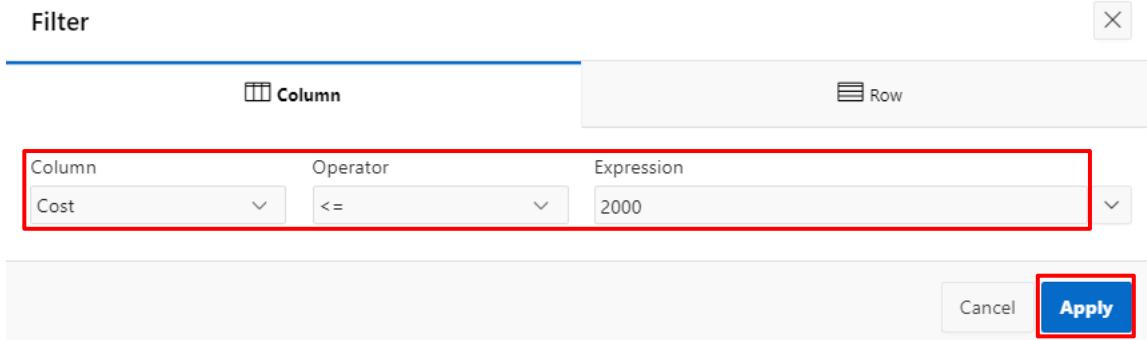
SQL Source

Home App 115619 Page 1 Session Debug Quick Edit Customize Help

b. Under Filter, enter/select the following:

- Column: **Cost**
- Operator: **<=**
- Expression: **2000**

Then, click **Apply**.



c. Next, add the second filter on the **Project** column. Click the **Actions** menu and select **Filter**.

Interactive Report

The screenshot shows an 'Interactive Report' interface. A filter bar at the top left shows 'Cost <= 2000' with a checked checkbox. On the right, the 'Actions' menu is open, with the 'Filter' option highlighted with a red box. The main area displays a table of tasks across various projects, with columns for Project, Task Name, Status, Cost, Budget, and Available Budget. The table includes rows for Bug Tracker, Discussion Forum, Public Website, and Employee Satisfaction Survey tasks.

d. Under Filter, select the following:

- Column: **Project**
- Operator: **=**
- Expression: **Bug Tracker**

Click **Apply**.

Filter

| Column | Operator | Expression |
|---------|----------|-------------|
| Project | = | Bug Tracker |

Cancel **Apply**

| Project | Task Name | Assigned To | Status | Cost | Budget | Available Budget |
|-------------|--------------------------------------|---------------|---------|-------|--------|------------------|
| Bug Tracker | Review automated testing tools | Myra Sutcliff | Closed | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | Myra Sutcliff | Closed | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | Myra Sutcliff | On-Hold | 1,000 | 2,000 | 1,000 |
| Bug Tracker | Measure effectiveness of improved QA | Myra Sutcliff | Pending | 0 | 500 | 500 |
| | | | | 1,850 | 4,100 | |

- e. Notice that two filters are applied. Fewer rows are displayed because only the rows that meet both the filter criteria are displayed. You can remove each filter by clicking the Remove Filter icon, next to the filter you want to remove.

Click the **Remove Filter** icon next to both the filters.

| Project | Task Name | Assigned To | Status | Cost | Budget | Available Budget |
|-------------|--------------------------------------|---------------|---------|-------|--------|------------------|
| Bug Tracker | Review automated testing tools | Myra Sutcliff | Closed | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | Myra Sutcliff | Closed | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | Myra Sutcliff | On-Hold | 1,000 | 2,000 | 1,000 |
| Bug Tracker | Measure effectiveness of improved QA | Myra Sutcliff | Pending | 0 | 500 | 500 |
| | | | | 1,850 | 4,100 | |

1 - 4 of 4

6. Now imagine you want to sort the report on the **Cost** column.

a. Select **Actions > Data > Sort**.

The screenshot shows a table with columns: Project, Task Name, Status, Cost, Budget, and Available Budget. The Actions menu is open, and the 'Sort' option under the 'Data' section is highlighted.

| Project | Task Name | Status | Cost | Budget | Available Budget |
|--------------------------|---|-----------|-------|--------|------------------|
| Bug Tracker | Document quality assurance procedures | Closed | 3,000 | 2,000 | -1,000 |
| Bug Tracker | Review automated testing tools | 750 | 1,500 | 750 | |
| Bug Tracker | Implement bug tracking software | Aggregate | 100 | 100 | 0 |
| Discussion Forum | Identify owners | Compute | 250 | 300 | 50 |
| Maintain Support Systems | HR software upgrades | Flashback | 8,000 | 7,000 | -1,000 |
| Discussion Forum | Install ACME Web Express application on internet server | Closed | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | On-Hold | 1,000 | 2,000 | 1,000 |
| Maintain Support Systems | Apply Billing System updates | Closed | 9,500 | 7,000 | -2,500 |
| Public Website | Determine host server | Closed | 200 | 200 | 0 |
| Public Website | Check software licenses | Closed | 100 | 100 | 0 |

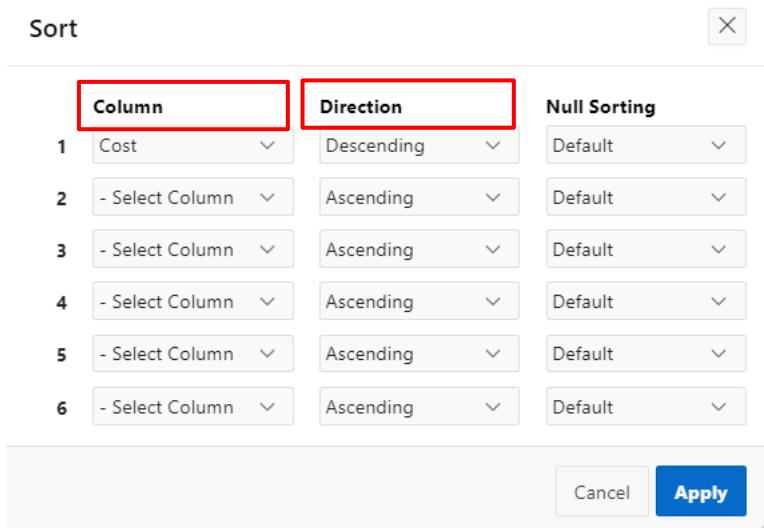
b. Remove the existing sort columns under **Column** by using the select list in rows 2 and 3 of Column to restore the value to the default, **Select Column**.

The 'Sort' dialog box contains six entries:

| Column | Direction | Null Sorting |
|-------------------|-----------|--------------|
| 1 Start Date | Ascending | Default |
| 2 End Date | Ascending | Default |
| 3 Project | Ascending | Default |
| 4 - Select Column | Ascending | Default |
| 5 - Select Column | Ascending | Default |
| 6 - Select Column | Ascending | Default |

At the bottom right are 'Cancel' and 'Apply' buttons.

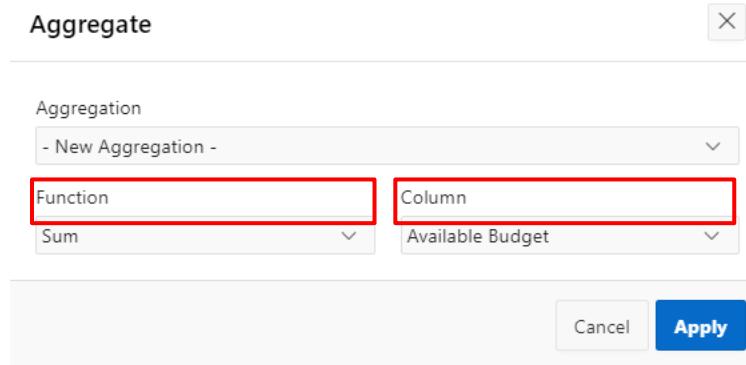
- c. Select **Cost** for Column, **Descending** for Direction, and click **Apply**.



- d. The report is now sorted in **Descending** order of **Cost**.
7. Create an aggregation against the **Available Budget** column. You want to display the sum of the **Available Budget**.
- a. Select **Actions > Data > Aggregate**.

| Project | Task Name | Status | Cost | Budget | Available Budget |
|--------------------------|--|--------|-------|--------|------------------|
| Maintain Support Systems | Apply Billing System updates | Closed | 9,500 | 7,000 | -2,500 |
| Maintain Support Systems | HR software upgrades | | 8,000 | 7,000 | -1,000 |
| Convert Spreadsheets | Create ACME Web Express applications from spreadsheets | | 6,000 | 10,000 | 4,000 |
| Client Server Conversion | Migrate pilot Client Server to ACME Web Express | | 4,500 | 5,000 | 500 |
| Bug Tracker | Document quality assurance procedures | | 3,000 | 2,000 | -1,000 |
| Email Integration | Complete plan | Closed | 3,000 | 1,500 | -1,500 |
| Convert Spreadsheets | Collect mission-critical spreadsheets | | 2,500 | 4,000 | 1,500 |
| Email Integration | Get RFPs for new server | Closed | 2,000 | 1,000 | -1,000 |
| Maintain Support Systems | Investigate new Virus Protection software | Open | 1,700 | 1,500 | -200 |
| Load Packaged Apps | Customize solutions | Open | 1,500 | 4,000 | 2,500 |

- b. In the **Aggregate** dialog box, select **Sum** for Function and **Available Budget** for Column. Click **Apply** to add the aggregation at the end of the last column of the report.



- c. The aggregate function is applied on the column. Notice that the sum of **Available Budget** is displayed at the end of the report under the column. Navigate until the last page to view the sum.

| Project | Task Name | Assigned To | Status | Cost | Budget | Available Budget |
|----------------------------|---|---------------|---------|--------|---------|------------------|
| Migrate from Legacy Server | Decommission Legacy Server | Al Bines | Pending | 0 | 500 | 500 |
| Convert Spreadsheets | Send links to previous spreadsheet owners | Pam King | Pending | 0 | 500 | 500 |
| Client Server Conversion | End-user Training | Myra Sutcliff | Pending | 0 | 2,500 | 2,500 |
| | | | | 59,000 | 130,450 | 71,450 |

8. In the report, assume you want to include **Cost** calculated with tax. The computation you want to make is **Cost * 1.05**. You create a computed column in the interactive report.

- a. Select **Actions > Data > Compute**.

| Project | Task Name | Assigned To | Status | Cost | Budget | Available Budget |
|------------------------------|------------------------------------|-------------|--------------|------|--------|------------------|
| Migrate from Legacy Server | Identify integration points | Bob | Filter | 0 | 2,000 | 2,000 |
| Migrate from Legacy Server | Create DB Connection to new server | Sam | Data | 0 | 100 | 100 |
| Migrate from Legacy Server | Migrate table structures | John | Format | 0 | 2,500 | 2,500 |
| Migrate from Legacy Server | Import data | John | Chart | 0 | 1,000 | 1,000 |
| Migrate from Legacy Server | Convert processes | Pam | Group By | 0 | 3,000 | 3,000 |
| Migrate from Legacy Server | Notify users | Bob | Pivot | 0 | 200 | 200 |
| Migrate from Legacy Server | Cut over to new database | Bob | Report | 0 | 1,500 | 1,500 |
| Client Server Conversion | Test migrated applications | Russ | Download | 0 | 6,000 | 6,000 |
| Employee Satisfaction Survey | Plan rollout schedule | Irene | Subscription | 0 | 750 | 750 |
| Email Integration | Purchase backup server | Bob | Help | 0 | 3,000 | 3,000 |

- b. The Compute dialog box appears.
- For Column Label, enter **Cost with Tax**.
 - For Computation Expression, click **Cost** under Column Aliases. Click *1.05 in the Keypad.

Click **Apply**.

The screenshot shows the 'Compute' dialog box. At the top, there's a 'Computation' section with a dropdown menu showing '- New Computation -'. Below it, the 'Column Label' field contains 'Cost with Tax', which is highlighted with a red box. To the right is a 'Format Mask' dropdown. Under 'Computation Expression', the formula 'H *1.05' is displayed. A green circular icon with a 'G' and a gear symbol is positioned next to the formula. Below the expression is a '► Examples' button. On the left, a 'Column Aliases' list includes items like D. Start Date, E. End Date, F. Status, G. Assigned To, H. Cost (which is highlighted with a red box), I. Budget, and K. Available Budget. In the center is a 'Keypad' with standard arithmetic operators. On the right, a 'Functions / Operators' list includes comparison operators (!=, <, <=, =, >, >=) and a 'ABS' function, with the multiplication operator '*' highlighted with a red box. At the bottom right are 'Cancel' and 'Apply' buttons, with 'Apply' being the active button.

- c. The new computed column **Cost with Tax** now appears in the report.

The screenshot shows a report titled "1. Primary Report" with various columns: Project, Task Name, Assigned To, Status, Cost, Budget, Available Budget, and Cost with Tax. The "Cost with Tax" column is highlighted with a red border. The data includes tasks like "Maintain Support Systems", "Convert Spreadsheets", and "Load Packaged Apps", along with their respective costs and statuses.

| Project | Task Name | Assigned To | Status | Cost | Budget | Available Budget | Cost with Tax |
|--------------------------|--|---------------|--------|-------|--------|------------------|---------------|
| Maintain Support Systems | Apply Billing System updates | Russ Saunders | Closed | 9,500 | 7,000 | -2,500 | 9975 |
| Maintain Support Systems | HR software upgrades | Pam King | Closed | 8,000 | 7,000 | -1,000 | 8400 |
| Convert Spreadsheets | Create ACME Web Express applications from spreadsheets | Pam King | Open | 6,000 | 10,000 | 4,000 | 6300 |
| Client Server Conversion | Migrate pilot Client Server to ACME Web Express | Scott Spencer | Closed | 4,500 | 5,000 | 500 | 4725 |
| Bug Tracker | Document quality assurance procedures | Myra Sutcliff | Closed | 3,000 | 2,000 | -1,000 | 3150 |
| Email Integration | Complete plan | Bob Nile | Closed | 3,000 | 1,500 | -1,500 | 3150 |
| Convert Spreadsheets | Collect mission-critical spreadsheets | Pam King | Closed | 2,500 | 4,000 | 1,500 | 2625 |
| Email Integration | Get RFPs for new server | Bob Nile | Closed | 2,000 | 1,000 | -1,000 | 2100 |
| Maintain Support Systems | Investigate new Virus Protection software | Pam King | Open | 1,700 | 1,500 | -200 | 1785 |
| Load Packaged Apps | Customize solutions | John Watson | Open | 1,500 | 4,000 | 2,500 | 1575 |

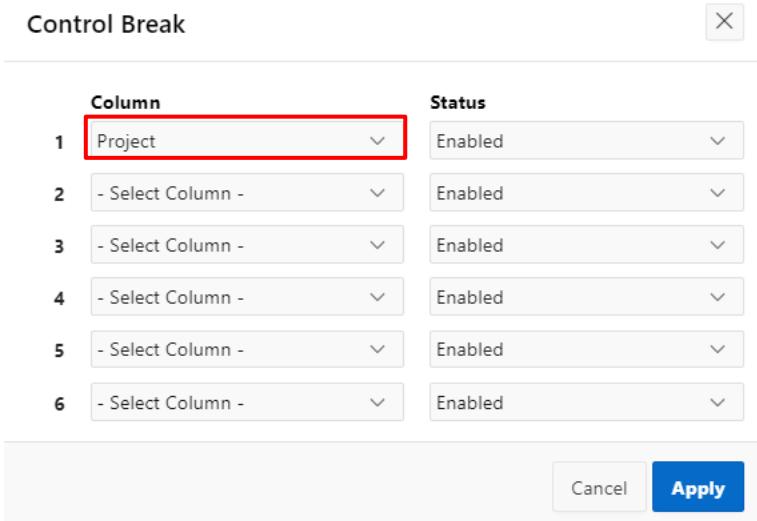
9. Create a control break on the **Project** column.

- a. Select **Actions > Format > Control Break**.

The screenshot shows the same report with the "Format" context menu open over the "Project" column. The "Control Break" option is highlighted in blue. Other options visible in the menu include Columns, Filter, Data, Format, Chart, Group By, Pivot, Report, Download, Subscription, and Help.

| Project | Task Name | Assigned To | Status | Cost | Budget | Available Budget | Cost with Tax |
|--------------------------|--|---------------|--------|-------|--------|------------------|---------------|
| Maintain Support Systems | Apply Billing System updates | Russ Saunders | Closed | 9,500 | 7,000 | -2,500 | 9975 |
| Maintain Support Systems | HR software upgrades | Pam King | Closed | 8,000 | 7,000 | -1,000 | 8400 |
| Convert Spreadsheets | Create ACME Web Express applications from spreadsheets | Pam King | Open | 6,000 | 10,000 | 4,000 | 6300 |
| Client Server Conversion | Migrate pilot Client Server to ACME Web Express | Scott Spencer | Closed | 4,500 | 5,000 | 500 | 4725 |
| Bug Tracker | Document quality assurance procedures | Myra Sutcliff | Closed | 3,000 | 2,000 | -1,000 | 3150 |
| Email Integration | Complete plan | Bob Nile | Closed | 3,000 | 1,500 | -1,500 | 3150 |
| Convert Spreadsheets | Collect mission-critical spreadsheets | Pam King | Closed | 2,500 | 4,000 | 1,500 | 2625 |
| Email Integration | Get RFPs for new server | Bob Nile | Closed | 2,000 | 1,000 | -1,000 | 2100 |
| Maintain Support Systems | Investigate new Virus Protection software | Pam King | Open | 1,700 | 1,500 | -200 | 1785 |
| Load Packaged Apps | Customize solutions | John Watson | Open | 1,500 | 4,000 | 2,500 | 1575 |

- b. In the Control Break dialog box, in row number 1, select **Project** for Column and click **Apply**.



- c. The control break is now applied. Notice that the aggregation you created in a previous step appears at the end of each control break.

The screenshot shows an 'Interactive Report' window with two sections: 'Project: ACME Web Express Configuration' and 'Project: Bug Tracker'. Both sections have a header row with columns: Task Name, Assigned To, Status, Cost, Budget, Available Budget, and Cost with Tax. The 'Project: ACME Web Express Configuration' section contains 7 tasks. The 'Project: Bug Tracker' section contains 2 tasks. The 'Cost' column is sorted in descending order, indicated by an arrow icon.

| Task Name | Assigned To | Status | Cost | Budget | Available Budget | Cost with Tax |
|--|---------------|---------|-------|--------|------------------|---------------|
| Determine Web listener configuration(s) | James Cassidy | Closed | 600 | 500 | -100 | 630 |
| Run installation | James Cassidy | Closed | 200 | 200 | 0 | 210 |
| Identify server requirements | John Watson | Closed | 200 | 500 | 300 | 210 |
| Configure Workspace provisioning | John Watson | Open | 200 | 100 | -100 | 210 |
| Select servers for Development, Test, Production | James Cassidy | Open | 200 | 600 | 400 | 210 |
| Specify security authentication scheme(s) | John Watson | Open | 200 | 300 | 100 | 210 |
| Create pilot workspace | John Watson | Closed | 100 | 100 | 0 | 105 |
| | | | 1,700 | 2,300 | 600 | |
| Project: Bug Tracker | | | | | | |
| Document quality assurance procedures | Myra Sutcliff | Closed | 3,000 | 2,000 | -1,000 | 3150 |
| Train developers on tracking bugs | Myra Sutcliff | On-Hold | 1,000 | 2,000 | 1,000 | 1050 |

10. Let's say you want to highlight rows with a **Cost** value of less than or equal to **500**. You add highlights to rows while continuing with the control break that you created in the previous step.

- a. Select **Actions > Format > Highlight**.

Interactive Report

The screenshot shows an Oracle Interactive Report interface. A context menu is open over a table row, with the 'Highlight' option selected. The table has columns for Task Name, Assigned To, Status, Available Budget, and Cost with Tax. The 'Status' column contains various icons representing different task statuses. The 'Available Budget' and 'Cost with Tax' columns show numerical values.

- b. In the Highlight dialog box, for Name, enter **Cost less than or equals 500**. For Background Color, select **Green** and for Text Color, select **Black**. Under Highlight Condition, for Column, select **Cost** and for Operator, select **<=**. For Expression, enter **500**.

Click **Apply**.

The 'Highlight' dialog box is shown with several fields highlighted by a red border:

- Name:** Cost less than or equals 500
- Background Color:** #d0f1cc
- Text Color:** #000000
- Column:** Cost
- Operator:** <=
- Expression:** 500

The 'Apply' button at the bottom right is also highlighted with a red border.

- c. Notice that the rows that meet the condition are highlighted in Green color.

The screenshot shows an Oracle Interactive Report titled "Project: ACME Web Express Configuration". A filter bar at the top indicates "Highlight: Cost less than or equals 500". The report displays a table with columns: Task Name, Assigned To, Status, Cost, Budget, Available Budget, and Cost with Tax. One row, "Run installation", has its entire row highlighted in green, indicating it meets the cost threshold. The total values at the bottom are 1,700, 2,300, and 600 respectively.

| Task Name | Assigned To | Status | Cost | Budget | Available Budget | Cost with Tax |
|--|---------------|--------|-------|--------|------------------|---------------|
| Determine Web listener configuration(s) | James Cassidy | Closed | 600 | 500 | -100 | 630 |
| Run installation | James Cassidy | Closed | 200 | 200 | 0 | 210 |
| Identify server requirements | John Watson | Closed | 200 | 500 | 300 | 210 |
| Configure Workspace provisioning | John Watson | Open | 200 | 100 | -100 | 210 |
| Select servers for Development, Test, Production | James Cassidy | Open | 200 | 600 | 400 | 210 |
| Specify security authentication scheme(s) | John Watson | Open | 200 | 300 | 100 | 210 |
| Create pilot workspace | John Watson | Closed | 100 | 100 | 0 | 105 |
| | | | 1,700 | 2,300 | 600 | |

11. Now, you want to control the number of rows to display on your **Interactive Report** page.

- a. Select **Actions > Format > Rows Per Page > 5**.

The screenshot shows the same Oracle Interactive Report as before, but with a context menu open over the first few rows. The "Format" option is selected, and a submenu for "Rows Per Page" is shown, with the value "5" highlighted. This indicates that the report will now display only 5 rows per page.

- b. Notice that **Rows Per Page** in your interactive report is now **1-5**.

The screenshot shows the report after changing the "Rows Per Page" setting. Now, only 5 rows are visible on the page. At the bottom right, a page footer indicates "1 - 5 of 73", confirming that there are more pages available.

| Task Name | Assigned To | Status | Cost | Budget | Available Budget | Cost with Tax |
|--|---------------|--------|------|--------|------------------|---------------|
| Determine Web listener configuration(s) | James Cassidy | Closed | 600 | 500 | -100 | 630 |
| Run installation | James Cassidy | Closed | 200 | 200 | 0 | 210 |
| Identify server requirements | John Watson | Closed | 200 | 500 | 300 | 210 |
| Configure Workspace provisioning | John Watson | Open | 200 | 100 | -100 | 210 |
| Select servers for Development, Test, Production | James Cassidy | Open | 200 | 600 | 400 | 210 |

12. In your interactive report, you want to include a chart to display the average **Cost** for each **Project**. Your interactive report should include both the Report and Chart views to toggle.

- a. Select **Actions > Chart**.

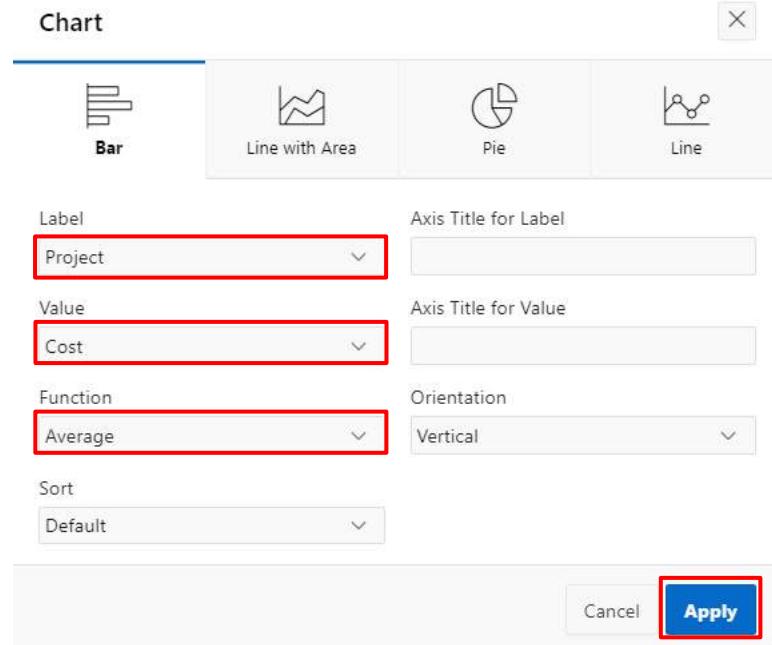
The screenshot shows a report titled "1. Primary Report" with a filter applied: "Control Break: Project" and "Highlight: Cost less than or equals". The table has columns: Task Name, Assigned To, Status, Budget, Available Budget, and Cost with Tax. The last row of the table has a context menu open, with the "Chart" option highlighted. Other options in the menu include Columns, Filter, Data, Format, Group By, Pivot, Report, Download, Subscription, and Help.

| Task Name | Assigned To | Status | Budget | Available Budget | Cost with Tax |
|--|---------------|--------|--------|------------------|---------------|
| Determine Web listener configuration(s) | James Cassidy | Closed | 500 | 500 | -100 |
| Run installation | James Cassidy | Closed | 100 | 200 | 0 |
| Identify server requirements | John Watson | Closed | 500 | 500 | 210 |
| Configure Workspace provisioning | John Watson | Open | 100 | 100 | 210 |
| Select servers for Development, Test, Production | James Cassidy | Open | 100 | 600 | 210 |
| Specify security authentication scheme(s) | John Watson | Open | 100 | 300 | 210 |
| Create pilot workspace | John Watson | Closed | 100 | 100 | 0 |
| | | | 1,700 | 2,300 | 600 |

- b. In the Chart dialog box, select/enter the following:

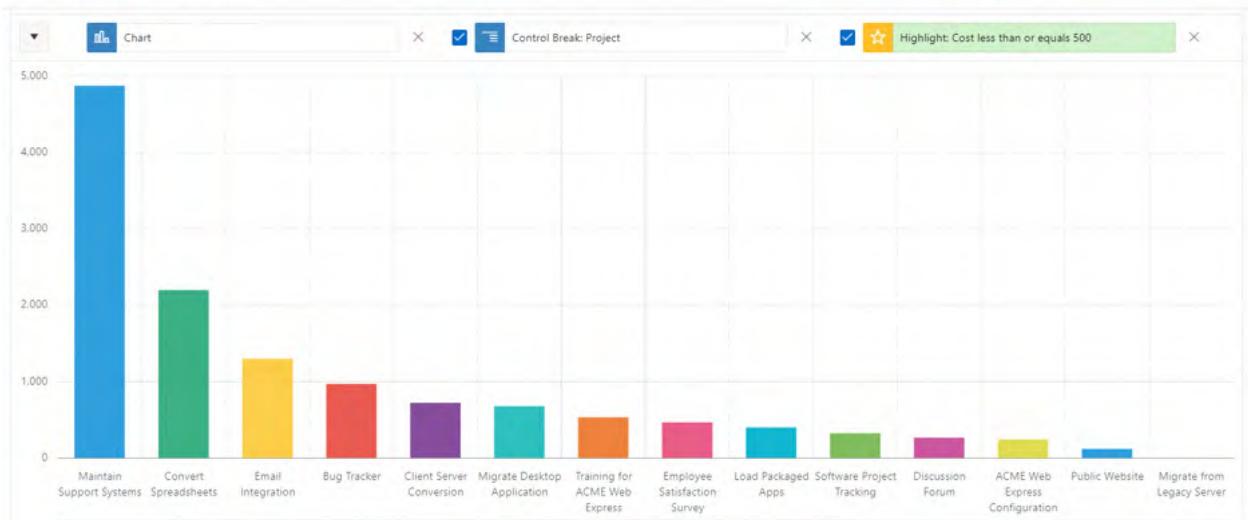
- Chart Type: **Bar**
- Label: **Project**
- Value: **Cost**
- Function: **Average**

Click **Apply**.



- c. The chart is created. Toggle between the Chart view and the Report view.

Interactive Report



13. Create a Group By report to display each **Project** with the **Total Budget**.

- Click **View Report**.
- Select **Actions > Group By**.

| Task Name | Assigned To | Status | Budget | Available Budget | Cost with Tax |
|--|---------------|--------|--------|------------------|---------------|
| Determine Web listener configuration(s) | James Cassidy | Closed | 500 | -100 | 630 |
| Run installation | James Cassidy | Closed | 200 | 0 | 210 |
| Identify server requirements | John Watson | Closed | 500 | 300 | 210 |
| Configure Workspace provisioning | John Watson | Open | 100 | -100 | 210 |
| Select servers for Development, Test, Production | James Cassidy | Open | 600 | 400 | 210 |
| Specify security authentication scheme(s) | John Watson | Open | 300 | 100 | 210 |
| Create pilot workspace | John Watson | Closed | 100 | 0 | 105 |
| | | | 2,300 | 600 | |

c. In the Group By dialog box, enter/select the following:

- Group By Column: **Project**
- Function: **Sum**
- Column: **Budget**
- Label: **Total Budget**
- Format Mask: **\$5,324.10**

Make sure you enable Sum and click **Apply**.

Group By

1 Project

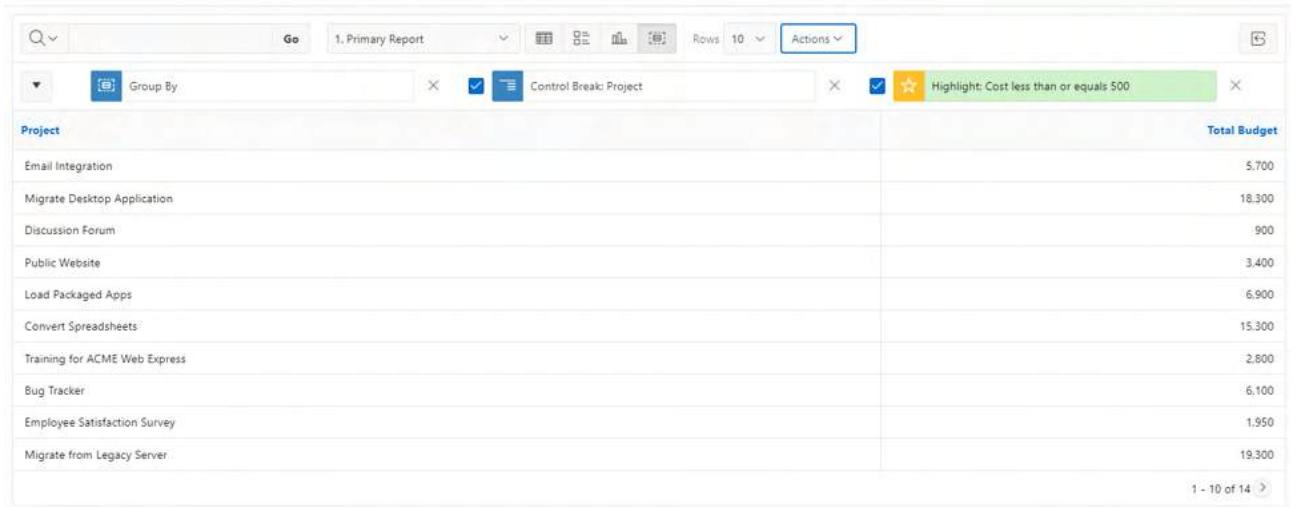
Add Group By Column

| | | | | |
|-----------|--------|--------------|-------------|-------------------------------------|
| Functions | Column | Label | Format Mask | Sum |
| 1 Sum | Budget | Total Budget | | <input checked="" type="checkbox"/> |

Add Function

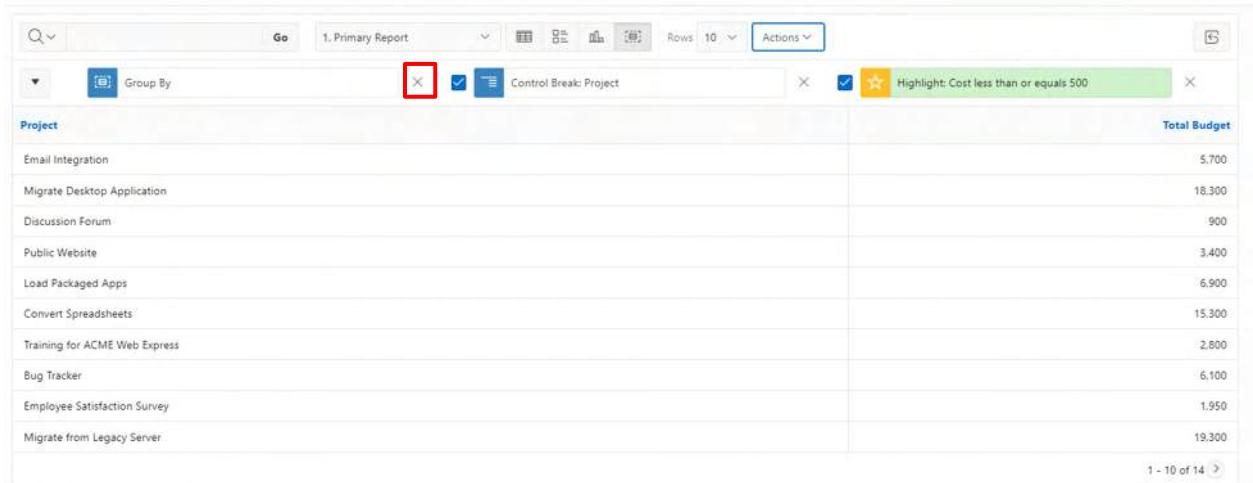
Cancel Apply

- d. The Group By report is created. You also see the sum of the budget. Notice that the icon for View Group By is also added.



| Project | Total Budget |
|-------------------------------|--------------|
| Email Integration | 5,700 |
| Migrate Desktop Application | 18,300 |
| Discussion Forum | 900 |
| Public Website | 3,400 |
| Load Packaged Apps | 6,900 |
| Convert Spreadsheets | 15,300 |
| Training for ACME Web Express | 2,800 |
| Bug Tracker | 6,100 |
| Employee Satisfaction Survey | 1,950 |
| Migrate from Legacy Server | 19,300 |

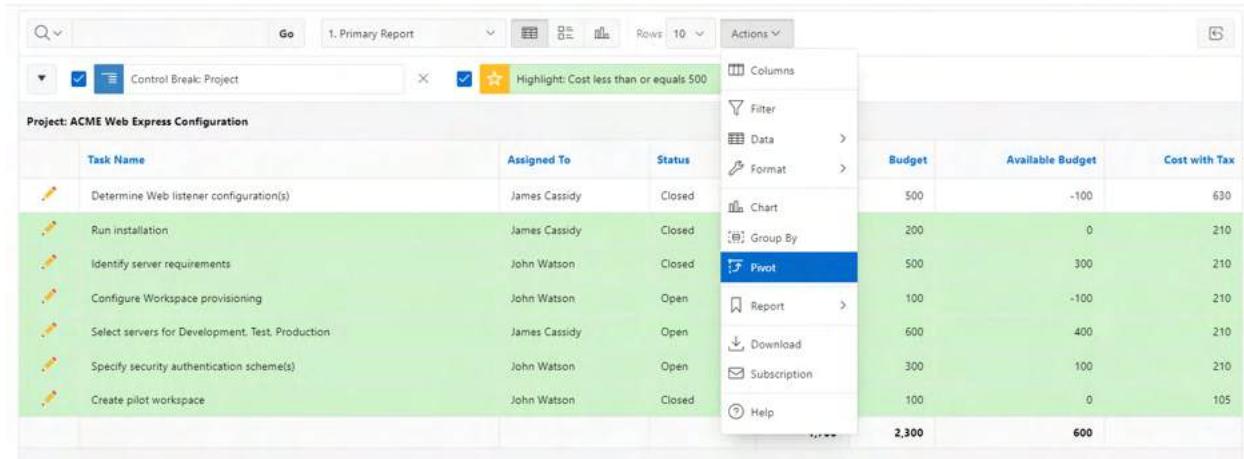
- e. Click the X to the right of **Edit Group By** to remove the filter.



| Project | Total Budget |
|-------------------------------|--------------|
| Email Integration | 5,700 |
| Migrate Desktop Application | 18,300 |
| Discussion Forum | 900 |
| Public Website | 3,400 |
| Load Packaged Apps | 6,900 |
| Convert Spreadsheets | 15,300 |
| Training for ACME Web Express | 2,800 |
| Bug Tracker | 6,100 |
| Employee Satisfaction Survey | 1,950 |
| Migrate from Legacy Server | 19,300 |

14. You want to display the **Sum** of Cost that is available with each **Project** and **Assigned To**.
 The results should be in a crosstab format. Create a Pivot report.

a. Click **Actions > Pivot**.



| Task Name | Assigned To | Status | Budget | Available Budget | Cost with Tax |
|--|---------------|--------|--------------|------------------|---------------|
| Determine Web listener configuration(s) | James Cassidy | Closed | 500 | -100 | 630 |
| Run installation | James Cassidy | Closed | 200 | 0 | 210 |
| Identify server requirements | John Watson | Closed | 500 | 300 | 210 |
| Configure Workspace provisioning | John Watson | Open | 100 | -100 | 210 |
| Select servers for Development, Test, Production | James Cassidy | Open | 600 | 400 | 210 |
| Specify security authentication scheme(s) | John Watson | Open | 300 | 100 | 210 |
| Create pilot workspace | John Watson | Closed | 100 | 0 | 105 |
| | | | 2,300 | 600 | |

b. In the Pivot dialog box, enter/select the following:

- Pivot Columns: **Assigned To**
- Row Columns: **Project**
- Functions: **Sum**
- Column: **Cost** and **Budget**
- Label: **Total Cost** and **Total Budget**
- Sum: **Yes**

Click **Apply**.

Pivot

Pivot Columns

1 Assigned To

Add Pivot Column

Row Columns

1 Project

Add Row Column

| Functions | Column | Label | Format Mask | Sum |
|-----------|--------|--------------|-------------|-------------------------------------|
| 1 Sum | Cost | Total Cost | | <input checked="" type="checkbox"/> |
| 2 Sum | Budget | Total Budget | | <input checked="" type="checkbox"/> |

Add Function

Cancel **Apply**

- c. The Pivot report is displayed, and a View Pivot icon is created next to the Actions button.

| All Bines | | Bob Nile | | Hank Davis | | Irene Jones | | James Cassidy | | John Watson | | Myra Sutcliff | | Pam King | | Russ Saunders | | Scott Spencer | | Tiger Scott | | |
|--------------------------------|------------|--------------|------------|--------------|------------|--------------|------------|---------------|------------|--------------|------------|---------------|------------|--------------|------------|---------------|------------|---------------|------------|--------------|------------|--------------|
| Project | Total Cost | Total Budget | Total Cost | Total Budget | Total Cost | Total Budget | Total Cost | Total Budget | Total Cost | Total Budget | Total Cost | Total Budget | Total Cost | Total Budget | Total Cost | Total Budget |
| ACME Web Express Configuration | - | - | - | - | - | - | - | - | 1,000 | 1,300 | 700 | 1,000 | - | - | - | - | - | - | - | - | - | |
| Bug Tracker | - | - | - | - | - | - | - | - | - | - | - | - | 4,850 | 6,100 | - | - | - | - | - | - | - | - |
| Client Server Conversion | - | - | - | - | - | - | - | - | - | - | - | - | 0 | 2,500 | 1,800 | 13,500 | 0 | 8,500 | 4,700 | 5,200 | - | - |
| Convert Spreadsheets | - | - | - | - | - | - | - | - | - | - | - | - | - | 8,800 | 15,300 | - | - | - | - | - | - | - |
| Discussion Forum | - | - | - | 800 | 900 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | |
| Email Integration | - | 5,200 | 5,700 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | |
| Employee Satisfaction Survey | - | - | - | - | - | - | - | 1,400 | 1,950 | - | - | - | - | - | - | - | - | - | - | - | - | |

15. You want to save the report with all the customizations.

- Select **Actions > Report > Save Report**.

Interactive Report

The screenshot shows an 'Interactive Report' interface. At the top, there's a toolbar with search, go, filter, data, format, rows (10), and actions buttons. A dropdown menu 'Control Break: Project' is open. Below the toolbar is a green header bar with the text 'Highlight: Cost less than or equals 500'. The main area is a grid of data with columns for Project, Total Cost, Total Budget, and various names like Al Bines, Bob Nile, Hank Davis, Irene Jones, James Cassidy, John Watson, and Myr. The grid has several rows of data. On the right side of the grid, there's a context menu with options: Report (selected), Save Report, Download, Subscription, Reset, and Help. The 'Save Report' option is highlighted with a blue border.

| Project | Total Cost | Total Budget |
|--------------------------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|
| ACME Web Express Configuration | - | - | - | - | - | - | - | - | 1,000 | 1,300 | 700 | 1,000 | - | - | - | - |
| Bug Tracker | - | - | - | - | - | - | - | - | - | - | - | - | 4,850 | - | - | - |
| Client Server Conversion | - | - | - | - | - | - | - | - | - | - | - | - | 0 | 1,500 | 0 | 8,500 |
| Convert Spreadsheets | - | - | - | - | - | - | - | - | - | - | - | - | 8,800 | 15,300 | - | - |
| Discussion Forum | - | - | - | - | 800 | 900 | - | - | - | - | - | - | - | - | - | - |
| Email Integration | - | - | 5,200 | 5,700 | - | - | - | - | - | - | - | - | - | - | - | - |
| Employee Satisfaction Survey | - | - | - | - | - | - | 1,400 | 1,950 | - | - | - | - | - | - | - | - |

- Under Name, enter **My Private Report** and click **Apply**.

The screenshot shows a 'Save Report' dialog box. It has a title bar with a close button. Inside, there's a section labeled 'Save (Only displayed for developers)' with a dropdown menu set to 'As Named Report'. Below that is a 'Name' field containing 'My Private Report', which is highlighted with a red box. There's also a 'Description' field and a 'Public' checkbox. At the bottom right are 'Cancel' and 'Apply' buttons, with 'Apply' being blue and bold.

Save (Only displayed for developers)
As Named Report

Name

Description

Public

Cancel **Apply**

16. A drop-down list automatically appears, with the report you just created being selected. You can view the default primary report. You want to reset the **Primary Report** back to the default settings and remove any customizations that you have made so far.

- Select **Primary Report** from the Reports drop-down list. The primary report is now displayed. You can make any changes to this report, and it will not be reflected in the **My Private Report** private report you just created.

The screenshot shows a report interface with a context menu open over a saved report named "My Private Report". The menu items are:

- 1. Primary Report
- 2. Budget Review
- 3. Highlighted Over Budget
- 4. Pivot Example
- Private
- 1. My Private Report
- Public

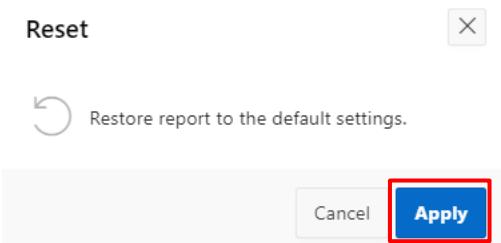
The report itself displays data for various projects like ACME Web Express Configuration, Bug Tracker, Client Server Conversion, etc., across different dimensions such as AI Bines, Bob Nile, Hank Davis, Irene Jones, James Cassidy, John Watson, Myra Sutcliff, Pam King, Russ Saunders, Scott Spencer, and Tiger Scott. The data includes Total Cost and Total Budget.

- b. To restore the primary report to its default settings, select **Actions > Report > Reset**.

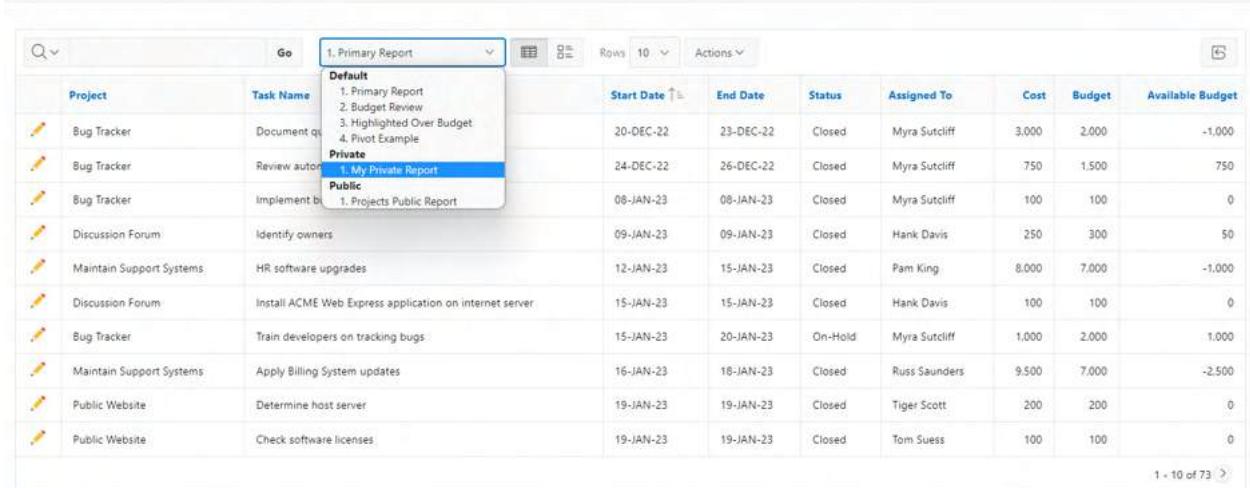
The screenshot shows the same report interface, but the Actions menu is open at the top right. The "Report" submenu is expanded, and the "Reset" option is highlighted with a blue box.

The report data remains the same, showing project details across various dimensions.

- c. In the Reset dialog box, click **Apply**.



- d. From the Reports drop-down list, select **My Private Report**.



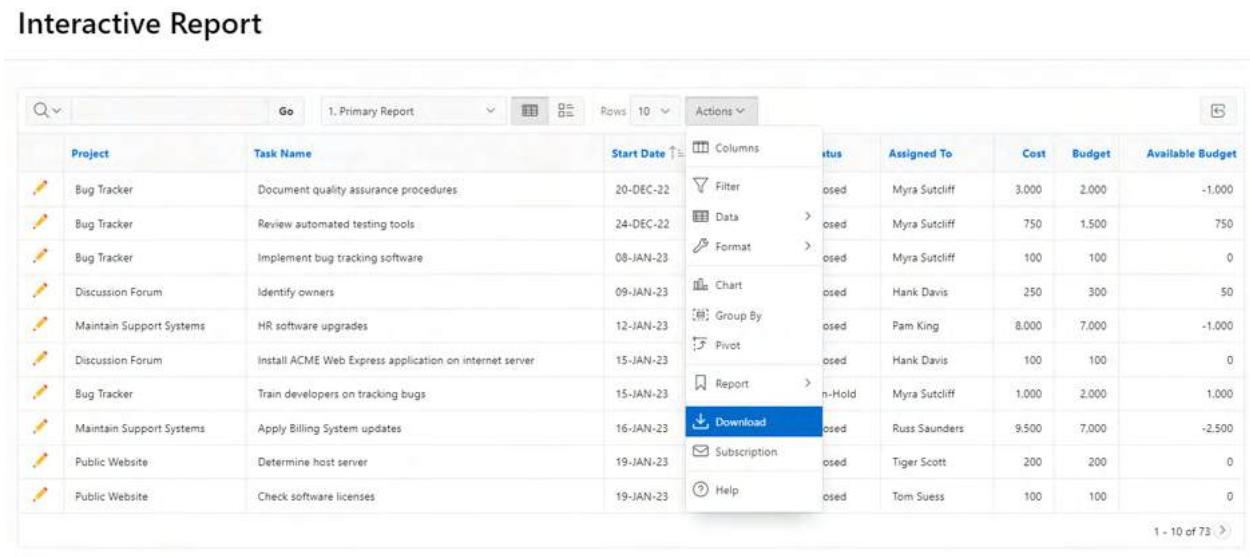
The screenshot shows a table of project tasks. The 'Reports' dropdown menu is open, and 'My Private Report' is selected. The table includes columns for Project, Task Name, Start Date, End Date, Status, Assigned To, Cost, Budget, and Available Budget. The data is as follows:

| Project | Task Name | Start Date ↑ | End Date | Status | Assigned To | Cost | Budget | Available Budget |
|--------------------------|---|--------------|-----------|---------|---------------|-------|--------|------------------|
| Bug Tracker | Document quality assurance procedures | 20-DEC-22 | 23-DEC-22 | Closed | Myra Sutcliff | 3,000 | 2,000 | -1,000 |
| Bug Tracker | Review automated testing tools | 24-DEC-22 | 26-DEC-22 | Closed | Myra Sutcliff | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | 08-JAN-23 | 08-JAN-23 | Closed | Myra Sutcliff | 100 | 100 | 0 |
| Discussion Forum | Identify owners | 09-JAN-23 | 09-JAN-23 | Closed | Hank Davis | 250 | 300 | 50 |
| Maintain Support Systems | HR software upgrades | 12-JAN-23 | 15-JAN-23 | Closed | Pam King | 8,000 | 7,000 | -1,000 |
| Discussion Forum | Install ACME Web Express application on internet server | 15-JAN-23 | 15-JAN-23 | Closed | Hank Davis | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | 15-JAN-23 | 20-JAN-23 | On-Hold | Myra Sutcliff | 1,000 | 2,000 | 1,000 |
| Maintain Support Systems | Apply Billing System updates | 16-JAN-23 | 18-JAN-23 | Closed | Russ Saunders | 9,500 | 7,000 | -2,500 |
| Public Website | Determine host server | 19-JAN-23 | 19-JAN-23 | Closed | Tiger Scott | 200 | 200 | 0 |
| Public Website | Check software licenses | 19-JAN-23 | 19-JAN-23 | Closed | Tom Suess | 100 | 100 | 0 |

17. You want to download the customized report as a CSV.

- a. Select **Actions > Download**.

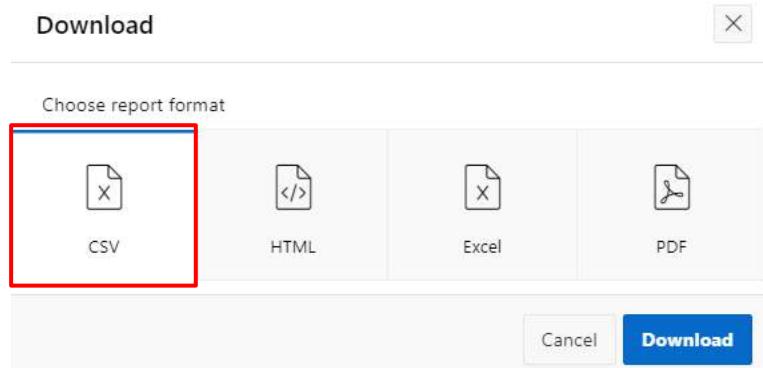
Interactive Report



The screenshot shows the same table of project tasks. The 'Actions' menu is open, and 'Download' is selected. The table structure is identical to the one above. The data is as follows:

| Project | Task Name | Start Date ↑ | Columns | Status | Assigned To | Cost | Budget | Available Budget |
|--------------------------|---|--------------|--------------|---------|---------------|-------|--------|------------------|
| Bug Tracker | Document quality assurance procedures | 20-DEC-22 | Filter | Closed | Myra Sutcliff | 3,000 | 2,000 | -1,000 |
| Bug Tracker | Review automated testing tools | 24-DEC-22 | Data | Closed | Myra Sutcliff | 750 | 1,500 | 750 |
| Bug Tracker | Implement bug tracking software | 08-JAN-23 | Format | Closed | Myra Sutcliff | 100 | 100 | 0 |
| Discussion Forum | Identify owners | 09-JAN-23 | Chart | Closed | Hank Davis | 250 | 300 | 50 |
| Maintain Support Systems | HR software upgrades | 12-JAN-23 | Group By | Closed | Pam King | 8,000 | 7,000 | -1,000 |
| Discussion Forum | Install ACME Web Express application on internet server | 15-JAN-23 | Pivot | Closed | Hank Davis | 100 | 100 | 0 |
| Bug Tracker | Train developers on tracking bugs | 15-JAN-23 | Report | On-Hold | Myra Sutcliff | 1,000 | 2,000 | 1,000 |
| Maintain Support Systems | Apply Billing System updates | 16-JAN-23 | Download | Closed | Russ Saunders | 9,500 | 7,000 | -2,500 |
| Public Website | Determine host server | 19-JAN-23 | Subscription | Closed | Tiger Scott | 200 | 200 | 0 |
| Public Website | Check software licenses | 19-JAN-23 | Help | Closed | Tom Suess | 100 | 100 | 0 |

- b. In the Download dialog box, select **CSV**.



- c. The report is now downloaded as a CSV.

| A | B | C | D | E | F | G | H | I |
|---------------------------------|---|------------|-----------|---------|---------------|-------|--------|------------------|
| Project | Task Name | Start Date | End Date | Status | Assigned To | Cost | Budget | Available Budget |
| 2 Bug Tracker | Document quality assurance procedures | 20-Dec-22 | 23-Dec-22 | Closed | Myra Sutcliff | 3,000 | 2,000 | -1,000 |
| 3 Bug Tracker | Review automated testing tools | 24-Dec-22 | 26-Dec-22 | Closed | Myra Sutcliff | 750 | 1,500 | 750 |
| 4 Bug Tracker | Implement bug tracking software | 8-Jan-23 | 8-Jan-23 | Closed | Myra Sutcliff | 100 | 100 | 0 |
| 5 Discussion Forum | Identify owners | 9-Jan-23 | 9-Jan-23 | Closed | Hank Davis | 250 | 300 | 50 |
| 6 Maintain Support Systems | HR software upgrades | 12-Jan-23 | 15-Jan-23 | Closed | Pam King | 8,000 | 7,000 | -1,000 |
| 7 Discussion Forum | Install ACME Web Express application on internet server | 15-Jan-23 | 15-Jan-23 | Closed | Hank Davis | 100 | 100 | 0 |
| 8 Bug Tracker | Train developers on tracking bugs | 15-Jan-23 | 20-Jan-23 | On-Hold | Myra Sutcliff | 1,000 | 2,000 | 1,000 |
| 9 Maintain Support Systems | Apply Billing System updates | 16-Jan-23 | 18-Jan-23 | Closed | Russ Saunders | 9,500 | 7,000 | -2,500 |
| 10 Public Website | Determine host server | 19-Jan-23 | 19-Jan-23 | Closed | Tiger Scott | 200 | 200 | 0 |
| 11 Public Website | Check software licenses | 19-Jan-23 | 19-Jan-23 | Closed | Tom Suess | 100 | 100 | 0 |
| 12 Employee Satisfaction Survey | Complete questionnaire | 19-Jan-23 | 20-Jan-23 | Closed | Irene Jones | 1,200 | 800 | -400 |
| 13 Maintain Support Systems | Arrange for vacation coverage | 20-Jan-23 | 20-Jan-23 | Open | Al Bines | 300 | 500 | 200 |
| 14 Public Website | Purchase additional software licenses, if needed | 20-Jan-23 | 21-Jan-23 | On-Hold | Al Bines | 300 | 1,000 | 700 |
| 15 Employee Satisfaction Survey | Review with legal | 21-Jan-23 | 21-Jan-23 | On-Hold | Irene Jones | 200 | 400 | 200 |
| 16 Employee Satisfaction Survey | Plan rollout schedule | 22-Jan-23 | 22-Jan-23 | On-Hold | Irene Jones | 0 | 750 | 750 |
| 17 Migrate Desktop Application | Identify pilot desktop applications | 24-Jan-23 | 24-Jan-23 | Closed | Bob Nile | 300 | 500 | 200 |
| 18 Email Integration | Complete plan | 26-Jan-23 | 27-Jan-23 | Closed | Bob Nile | 3,000 | 1,500 | -1,500 |
| 19 Migrate Desktop Application | Migrate pilot applications to ACME Web Express | 26-Jan-23 | 27-Jan-23 | Closed | Bob Nile | 1,250 | 1,500 | 250 |

You now know how to manage and customize an interactive report as an end user. You may now **proceed to the next practice**.

Practice: Managing and Customizing Interactive Grids

Practice 1: Customize an Interactive Grid as a Developer

Overview

In this practice, you customize the Interactive Grid page you created in the **Demo Projects** application as a **Developer**.

Downloads

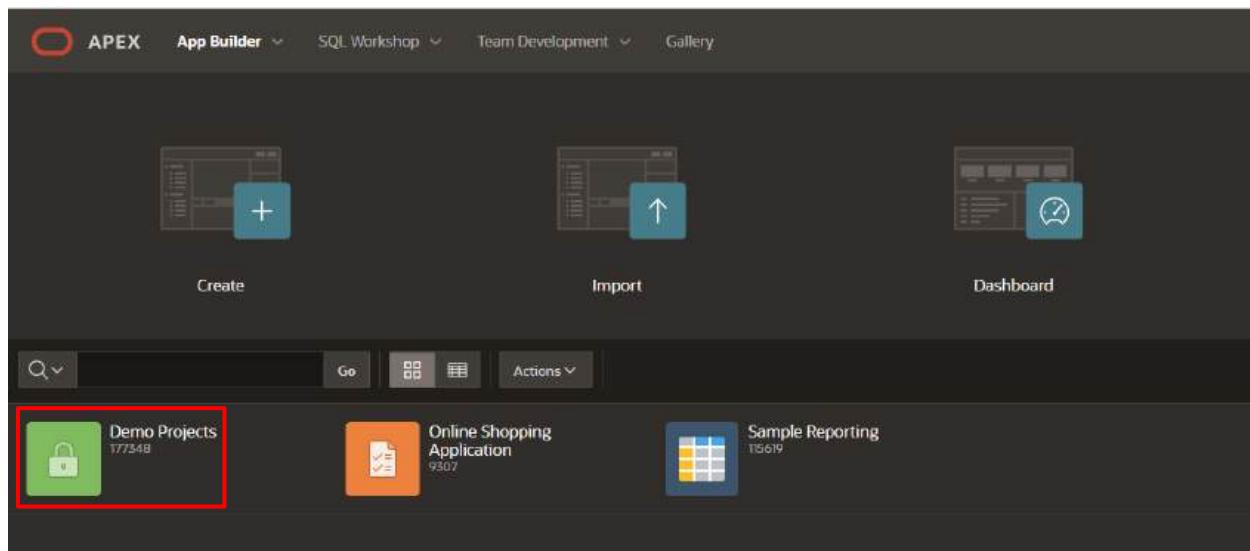
- Did you miss out on trying the previous practices? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, follow the steps described in **Practice-01** and **Practice-02**.

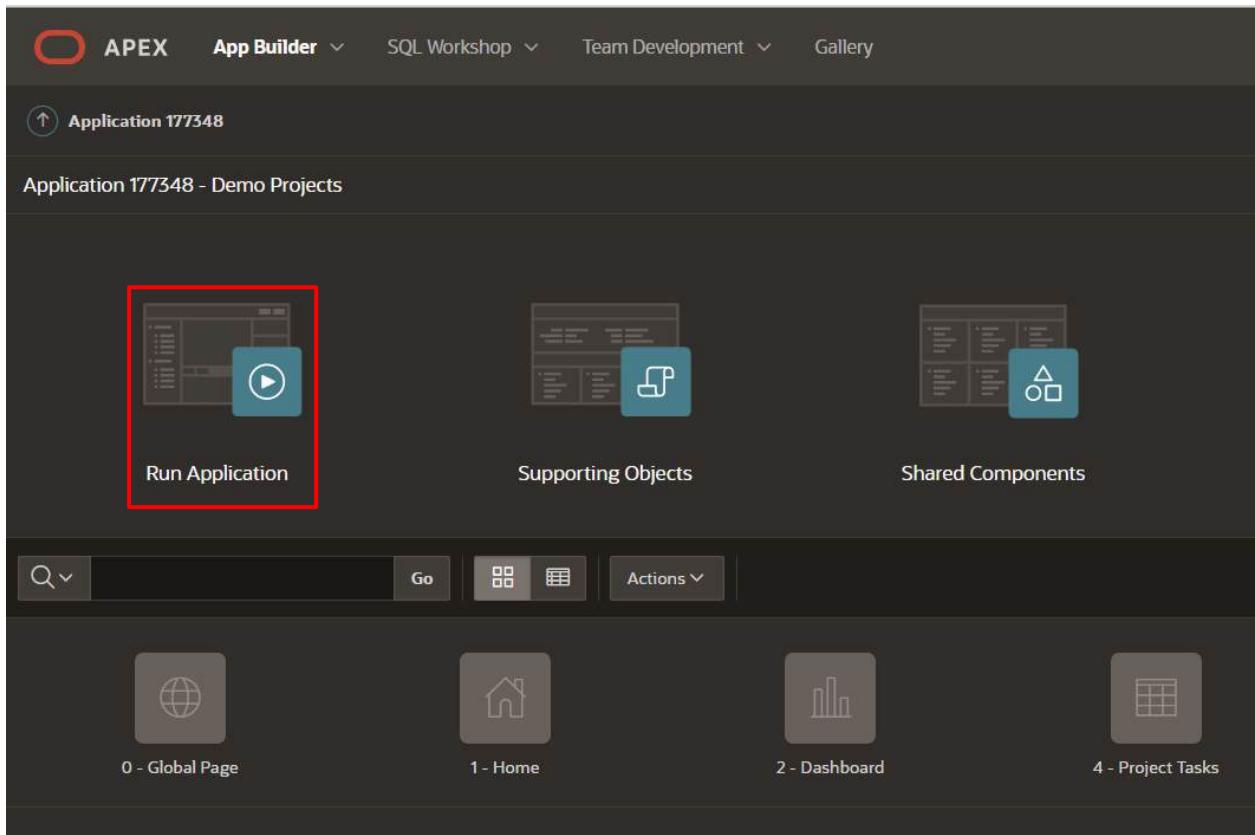
Tasks

Manage and Customize the Interactive Grid as a Developer

This practice uses the **Demo Projects** application. In this practice, you customize the **Interactive Grid** for end users. You create column groups, set the pagination type, and set the report downloadable formats that should be available to end users. You also enable end users to save the report as Public interactive grids and convert a read-only interactive grid to an editable interactive grid.

1. Navigate to **App Builder** and run the **Demo Projects** application.



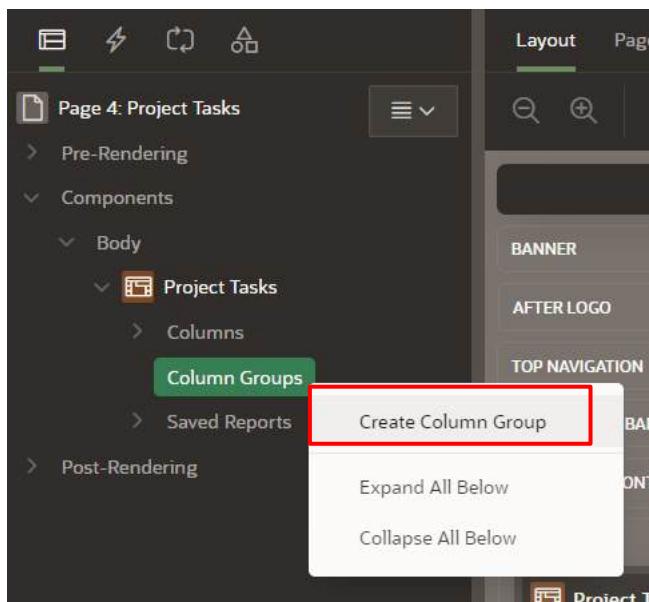


2. In the navigation menu, click **Project Tasks**. You want to customize the display of this interactive grid for your end users. In the Developer Toolbar, click **Edit Page 4**.

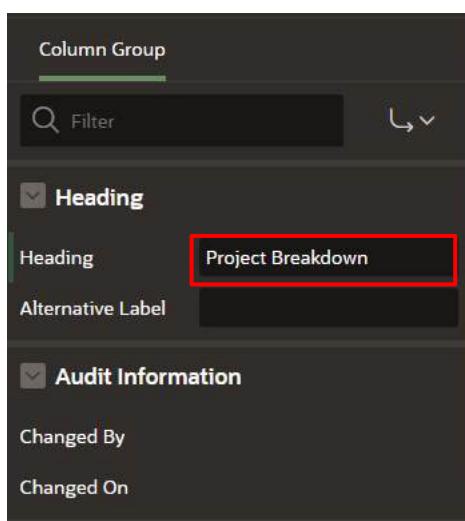
The screenshot shows the 'Project Tasks' page within the application. The left sidebar has links for Home, Dashboard, Project Tasks (which is highlighted with a red box), and Project Tasks Search. The main area displays a grid of project tasks with columns: Project, Task Name, Start Date, End Date, Status, Assigned To, Cost, and Budget. The grid contains 17 rows of task data. At the bottom, there's a developer toolbar with various buttons, and the 'Edit Page 4' button is also highlighted with a red box.

| Project | Task Name | Start Date | End Date | Status | Assigned To | Cost | Budget |
|---------------------------|--------------------------------|------------|------------|---------|---------------|------|--------|
| ACME Web Configuration | Identify server requiremen... | 11/19/2021 | 12/6/2021 | Closed | John Watson | 300 | 300 |
| Maintain Support Systems | HR software upgrades | 12/4/2021 | 1/1/2022 | On-Hold | Pam King | 8000 | 7000 |
| Maintain Support Systems | Apply Billing System upd... | 1/7/2022 | 1/19/2022 | On-Hold | Russ Sanders | 9500 | 7000 |
| ACME Web Configuration | Determine Web listener c... | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| ACME Web Configuration | Specify security authentic... | 1/2/2022 | 1/4/2022 | Closed | John Watson | 200 | 300 |
| ACME Web Configuration | Select servers for Develop... | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| Email Integration | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3000 | 1500 |
| ACME Web Configuration | Configure Workspace pro... | 12/20/2021 | 1/4/2022 | Closed | John Watson | 200 | 100 |
| ACME Web Configuration | Create pilot workspace | 1/15/2022 | 2/8/2022 | Closed | John Watson | 100 | 100 |
| ACME Web Configuration | Run installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| Bug Tracker | Implement bug tracking s... | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| Bug Tracker | Review automated testin... | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff | 2750 | 1500 |
| Train Developers | Publish development sta... | 12/3/2021 | 12/13/2021 | On-Hold | John Watson | 1000 | 2000 |
| Train Developers | Publish links to self-study... | 12/28/2021 | 1/3/2022 | Closed | John Watson | 100 | 100 |
| Train Developers | Create training workspace | 12/9/2021 | 12/11/2021 | Closed | James Cassidy | 500 | 700 |
| Load Packaged Application | Identify point solutions re... | 1/2/2022 | 1/24/2022 | Closed | John Watson | 200 | 300 |

3. Add column group headers to the interactive grid:
 - Project Breakdown: Project, Task_Name columns
 - Schedule: Start_Date, End_Date columns
 - Project Financing: Cost, Budget columns
 - a. In the Page Designer, under **Components > Body**, navigate to the **Project Tasks** Interactive Grid region and right-click **Column Groups**. Select **Create Column Group**.

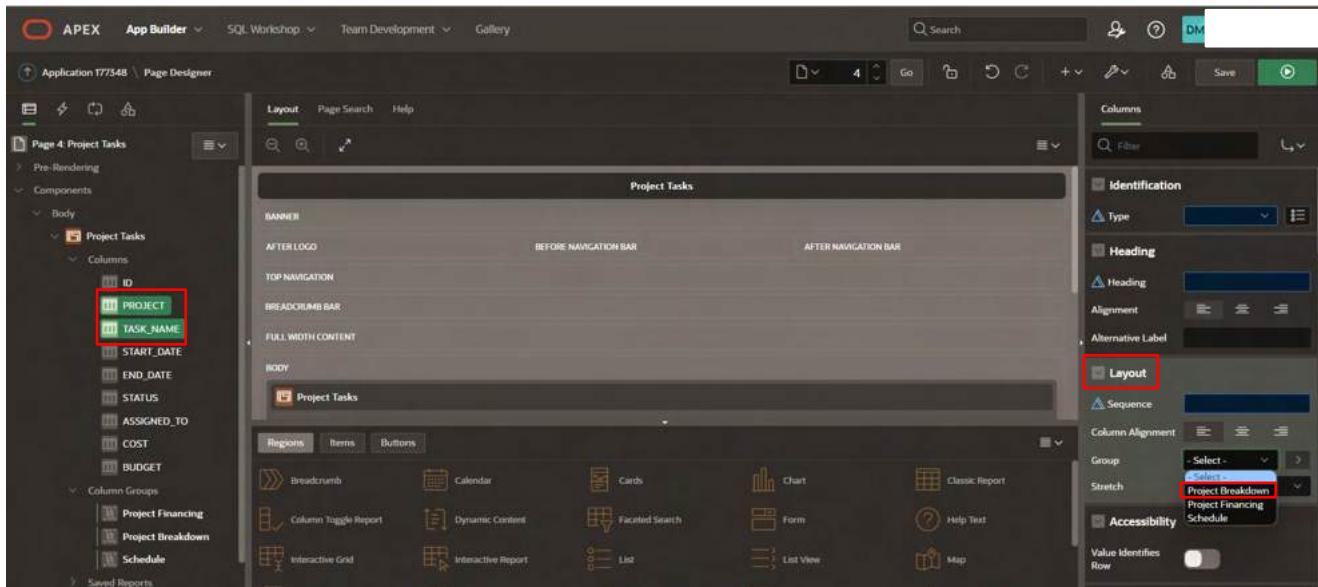


- b. In the Property Editor, for Heading, enter **Project Breakdown**.

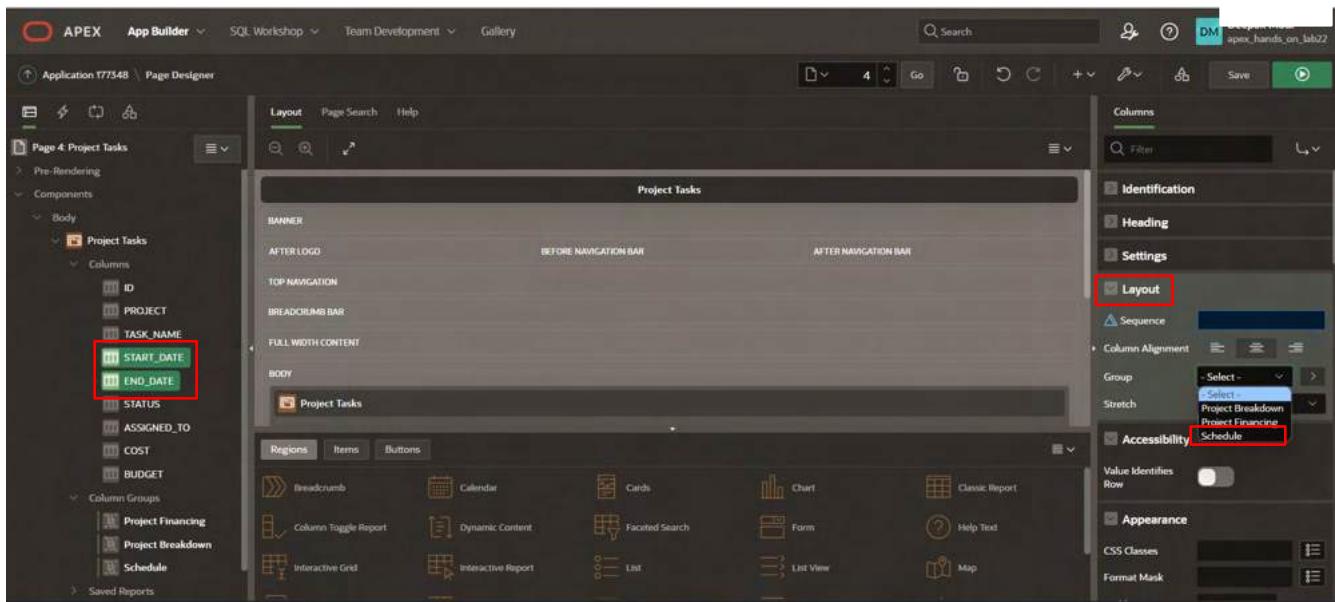


- c. Repeat steps **a** and **b** to create the **Schedule** and **Project Financing** column groups.

- d. Now that you have created column groups, you need to assign columns to them. Expand **Columns** and select the **Project** and **Task_Name** columns.
- e. In the Property Editor, under **Layout**, for Group, select **Project Breakdown**.



- f. Then, select the **StartDate** and **EndDate** columns. In the Property Editor, under **Layout**, for Group, select **Schedule**.



- g. Finally, select the **Cost** and **Budget** columns. In the Property Editor, under **Layout**, for Group, select **Project Financing**.

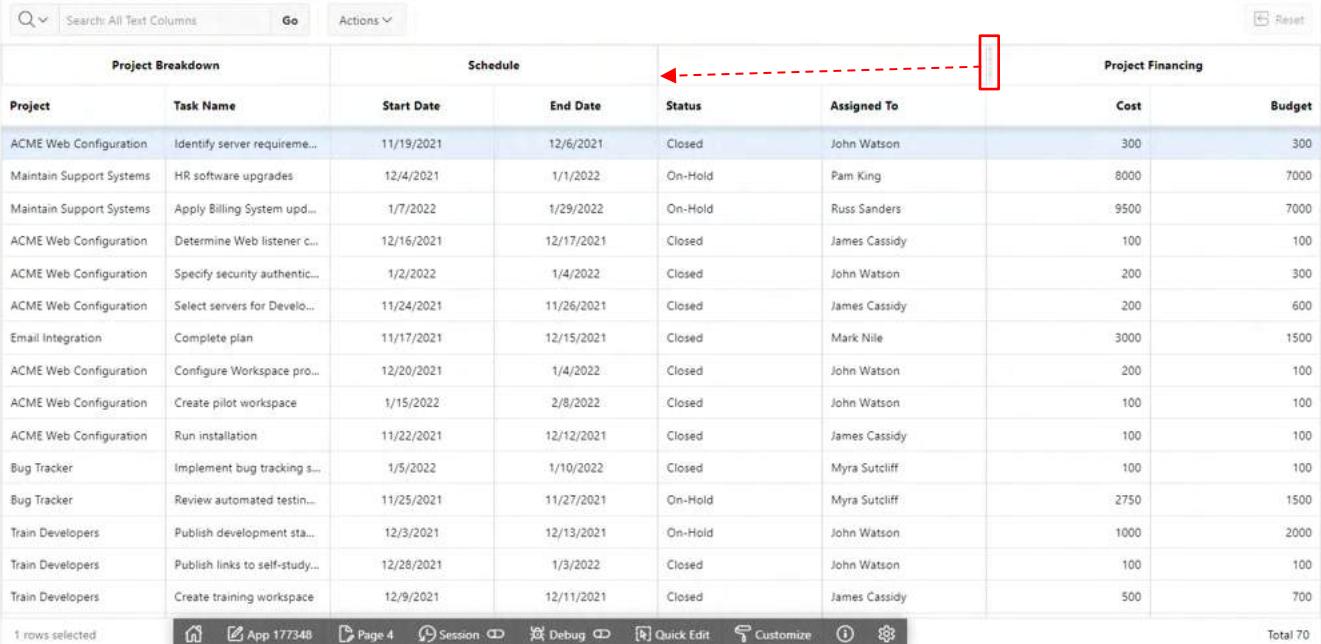
Then, click **Save** and **Run Page**.

h. The interactive grid now displays column groups.

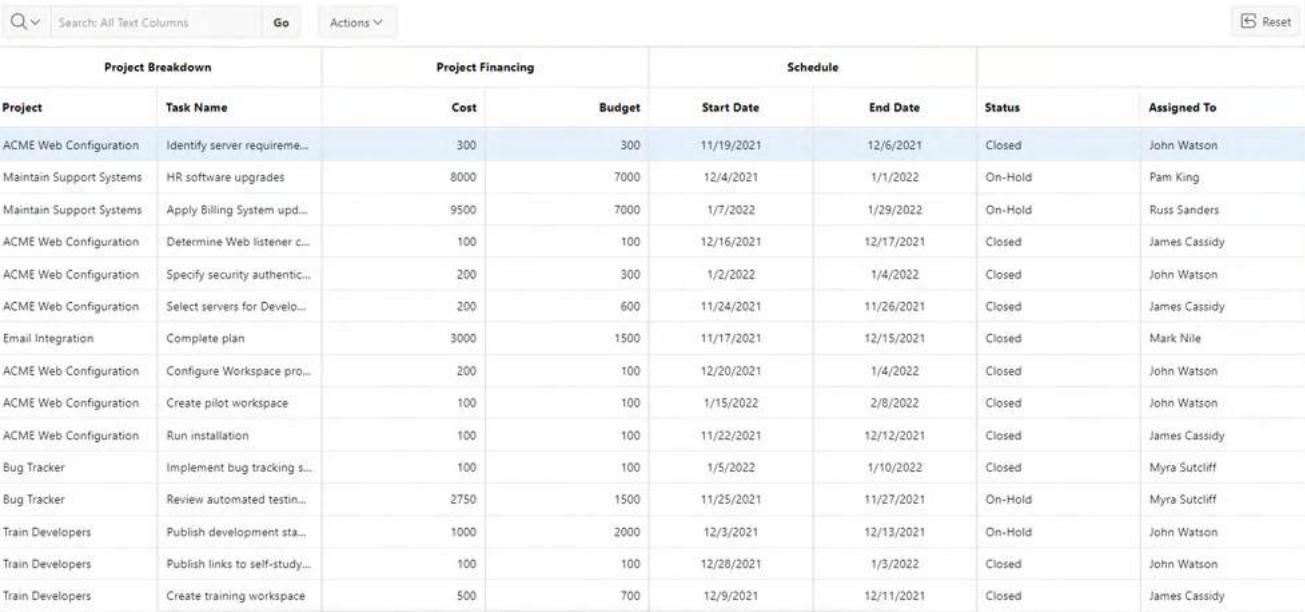
| Project Breakdown | | Schedule | | | | Project Financing | |
|--------------------------|--------------------------------|------------|------------|---------|---------------|-------------------|--------|
| Project | Task Name | Start Date | End Date | Status | Assigned To | Cost | Budget |
| ACME Web Configuration | Identify server requireme... | 11/19/2021 | 12/6/2021 | Closed | John Watson | 300 | 300 |
| Maintain Support Systems | HR software upgrades | 12/4/2021 | 1/1/2022 | On-Hold | Pam King | 8000 | 7000 |
| Maintain Support Systems | Apply Billing System upd... | 1/7/2022 | 1/29/2022 | On-Hold | Russ Sanders | 9500 | 7000 |
| ACME Web Configuration | Determine Web listener c... | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| ACME Web Configuration | Specify security authentic... | 1/2/2022 | 1/4/2022 | Closed | John Watson | 200 | 300 |
| ACME Web Configuration | Select servers for Develop... | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| Email Integration | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3000 | 1500 |
| ACME Web Configuration | Configure Workspace pro... | 12/20/2021 | 1/4/2022 | Closed | John Watson | 200 | 100 |
| ACME Web Configuration | Create pilot workspace | 1/15/2022 | 2/8/2022 | Closed | John Watson | 100 | 100 |
| ACME Web Configuration | Run installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| Bug Tracker | Implement bug tracking s... | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| Bug Tracker | Review automated testin... | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff | 2750 | 1500 |
| Train Developers | Publish development sta... | 12/3/2021 | 12/13/2021 | On-Hold | John Watson | 1000 | 2000 |
| Train Developers | Publish links to self-study... | 12/28/2021 | 1/3/2022 | Closed | John Watson | 100 | 100 |
| Train Developers | Create training workspace | 12/9/2021 | 12/11/2021 | Closed | James Cassidy | 500 | 700 |

- Rearrange the columns in the interactive grid. You want to ensure the Project Breakdown, Schedule, and Project Financing column groups display in order, followed by Status and Assigned To.
- Hover your mouse over the Project Financing column group header to display the drag handle. Your cursor also changes when it comes into contact with the drag handle. Click and hold the drag handle.

- b. Then, drag the column group to the Status column location. The heading shifts out of place in the row. The Project Financing column group should be followed by the Status column. Release the mouse. The Project Financing column group drops into place.

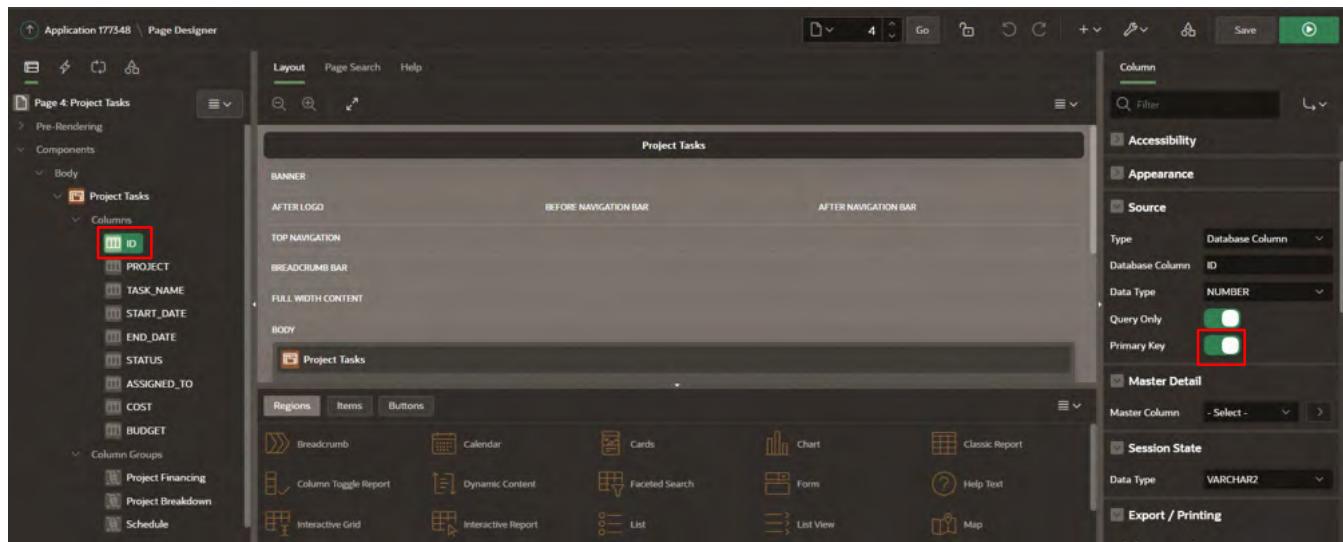


| Project Breakdown | | Schedule | | Status | Assigned To | Project Financing | |
|--------------------------|--------------------------------|------------|------------|---------|---------------|-------------------|--------|
| Project | Task Name | Start Date | End Date | | | Cost | Budget |
| ACME Web Configuration | Identify server requireme... | 11/19/2021 | 12/6/2021 | Closed | John Watson | 300 | 300 |
| Maintain Support Systems | HR software upgrades | 12/4/2021 | 1/1/2022 | On-Hold | Pam King | 8000 | 7000 |
| Maintain Support Systems | Apply Billing System upd... | 1/7/2022 | 1/29/2022 | On-Hold | Russ Sanders | 9500 | 7000 |
| ACME Web Configuration | Determine Web listener c... | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| ACME Web Configuration | Specify security authentic... | 1/2/2022 | 1/4/2022 | Closed | John Watson | 200 | 300 |
| ACME Web Configuration | Select servers for Develo... | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| Email Integration | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3000 | 1500 |
| ACME Web Configuration | Configure Workspace pro... | 12/20/2021 | 1/4/2022 | Closed | John Watson | 200 | 100 |
| ACME Web Configuration | Create pilot workspace | 1/15/2022 | 2/8/2022 | Closed | John Watson | 100 | 100 |
| ACME Web Configuration | Run installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| Bug Tracker | Implement bug tracking s... | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| Bug Tracker | Review automated testin... | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff | 2750 | 1500 |
| Train Developers | Publish development sta... | 12/3/2021 | 12/13/2021 | On-Hold | John Watson | 1000 | 2000 |
| Train Developers | Publish links to self-study... | 12/28/2021 | 1/3/2022 | Closed | John Watson | 100 | 100 |
| Train Developers | Create training workspace | 12/9/2021 | 12/11/2021 | Closed | James Cassidy | 500 | 700 |

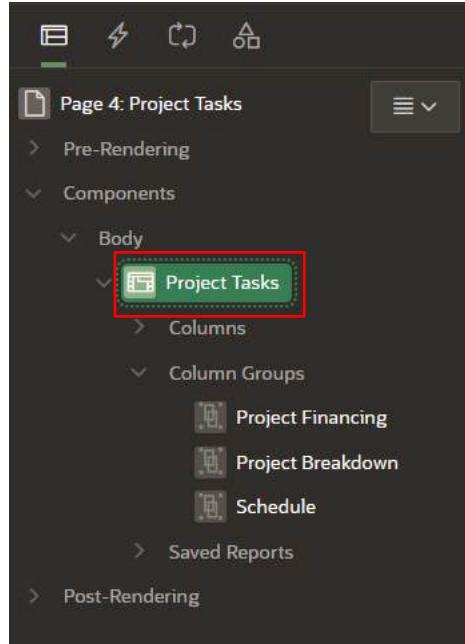


| Project Breakdown | | Project Financing | | Schedule | | | |
|--------------------------|--------------------------------|-------------------|--------|------------|------------|---------|---------------|
| Project | Task Name | Cost | Budget | Start Date | End Date | Status | Assigned To |
| ACME Web Configuration | Identify server requireme... | 300 | 300 | 11/19/2021 | 12/6/2021 | Closed | John Watson |
| Maintain Support Systems | HR software upgrades | 8000 | 7000 | 12/4/2021 | 1/1/2022 | On-Hold | Pam King |
| Maintain Support Systems | Apply Billing System upd... | 9500 | 7000 | 1/7/2022 | 1/29/2022 | On-Hold | Russ Sanders |
| ACME Web Configuration | Determine Web listener c... | 100 | 100 | 12/16/2021 | 12/17/2021 | Closed | James Cassidy |
| ACME Web Configuration | Specify security authentic... | 200 | 300 | 1/2/2022 | 1/4/2022 | Closed | John Watson |
| ACME Web Configuration | Select servers for Develo... | 200 | 600 | 11/24/2021 | 11/26/2021 | Closed | James Cassidy |
| Email Integration | Complete plan | 3000 | 1500 | 11/17/2021 | 12/15/2021 | Closed | Mark Nile |
| ACME Web Configuration | Configure Workspace pro... | 200 | 100 | 12/20/2021 | 1/4/2022 | Closed | John Watson |
| ACME Web Configuration | Create pilot workspace | 100 | 100 | 1/15/2022 | 2/8/2022 | Closed | John Watson |
| ACME Web Configuration | Run installation | 100 | 100 | 11/22/2021 | 12/12/2021 | Closed | James Cassidy |
| Bug Tracker | Implement bug tracking s... | 100 | 100 | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff |
| Bug Tracker | Review automated testin... | 2750 | 1500 | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff |
| Train Developers | Publish development sta... | 1000 | 2000 | 12/3/2021 | 12/13/2021 | On-Hold | John Watson |
| Train Developers | Publish links to self-study... | 100 | 100 | 12/28/2021 | 1/3/2022 | Closed | John Watson |
| Train Developers | Create training workspace | 500 | 700 | 12/9/2021 | 12/11/2021 | Closed | James Cassidy |

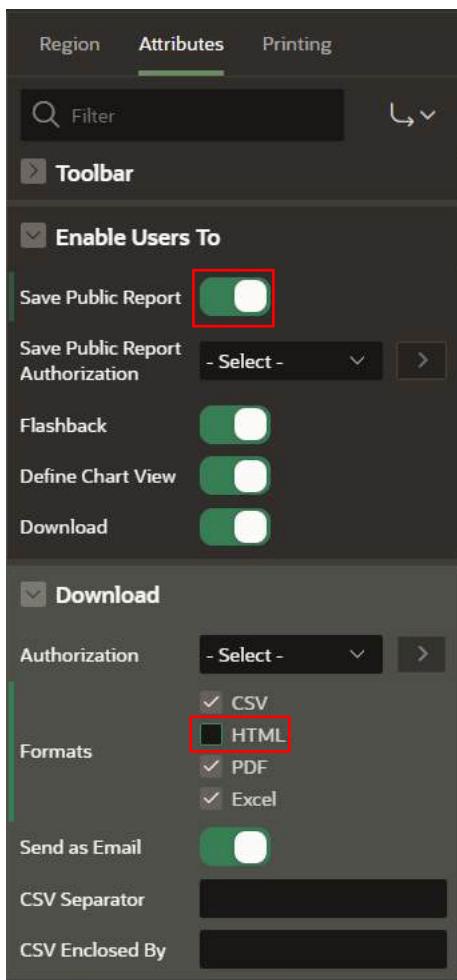
5. Click **Page n** in the runtime developer toolbar. You want to make the **ID** column as the **Primary Key**. This will help you make the interactive grid editable.



6. You want to ensure that end users are able to save Public interactive grids. You also want to exclude HTML from the download formats available to end users.
- Under Rendering, select the **Project Tasks** interactive grid region.

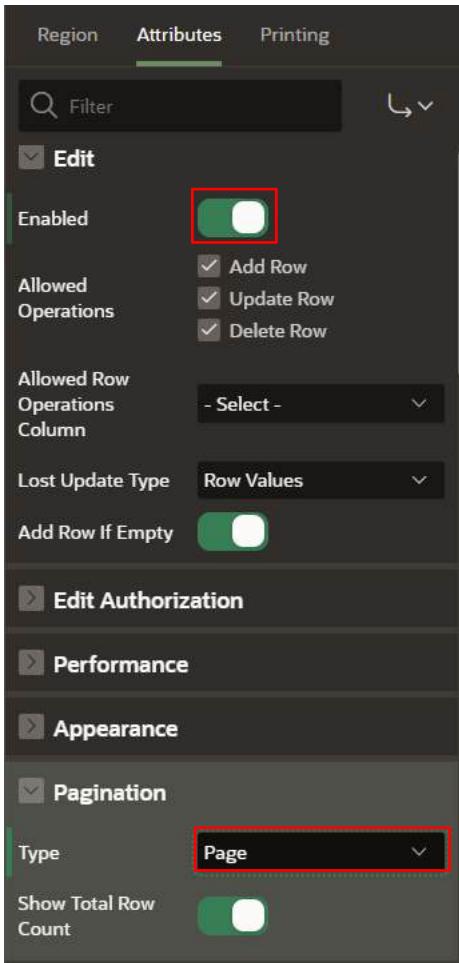


- b. In the Property Editor, select **Attributes** and then navigate to **Enable Users To**. Click **Save Public Report** to enable the feature. Under **Download**, deselect the **HTML** check box.

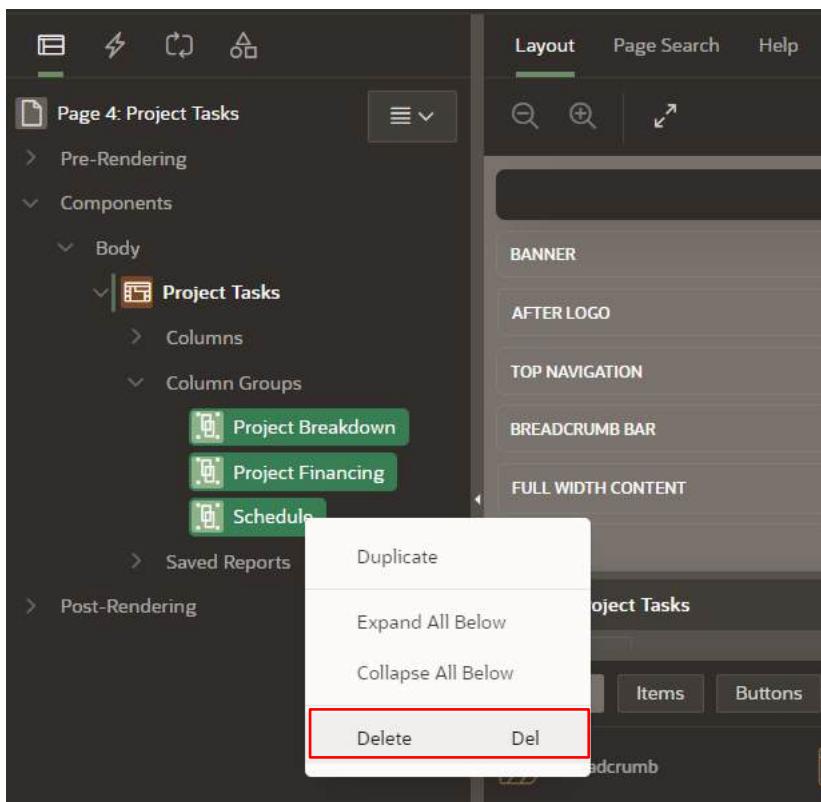


7. Convert this read-only interactive grid into an **editable interactive grid**. Then, reset the pagination as Page type, displaying the total row count.
- a. Under Rendering, select the **Project Tasks** interactive grid region.

- b. In the Property Editor, navigate to **Attributes** and then navigate to Edit. Click **Enabled** to turn on the feature. Also, under **Pagination**, for Type, select **Page**.

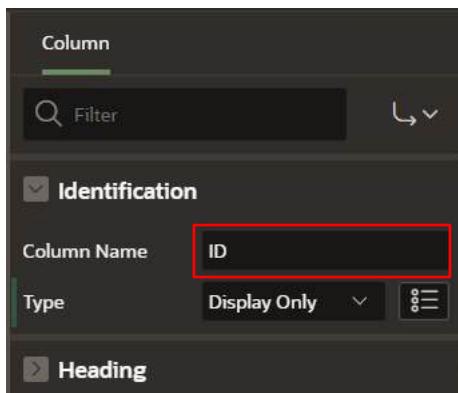


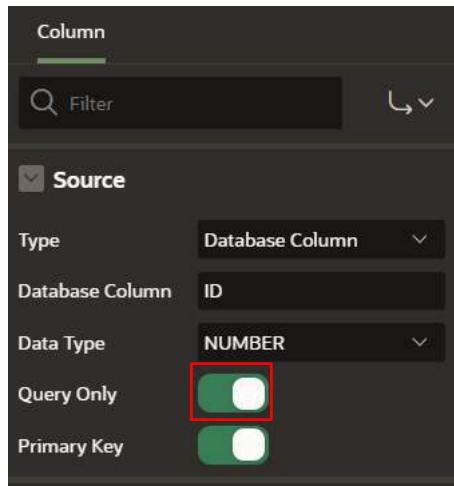
8. Delete the column groups in the interactive grid. Under **Rendering > Project Tasks** **Interactive Grid > Column Groups**, select **Schedule**, **Project Breakdown**, and **Project Financing**, right-click and select **Delete**.



9. Assume you want to display the ID column and exclude the ID column from DML operations. Under **Page Rendering > Project Tasks** interactive grid, expand Columns and select **ID**.

Navigate to **Identification** and set Type to **Display Only**. Then, navigate to **Source** and click **Query Only** to enable it. Click **Save** and **Run Page**.





You now know how to manage and customize the interactive grid as a developer. You may now **proceed to the next practice**.

Practice 2: Customize an Interactive Grid as an End User

Overview

In this practice, you customize the Interactive Grid page you created in the **Demo Projects** application as an **End User**.

Tasks

Customize an Interactive Grid as an End User

In this practice, you use and customize the display of your interactive grid. You also edit an editable interactive grid.

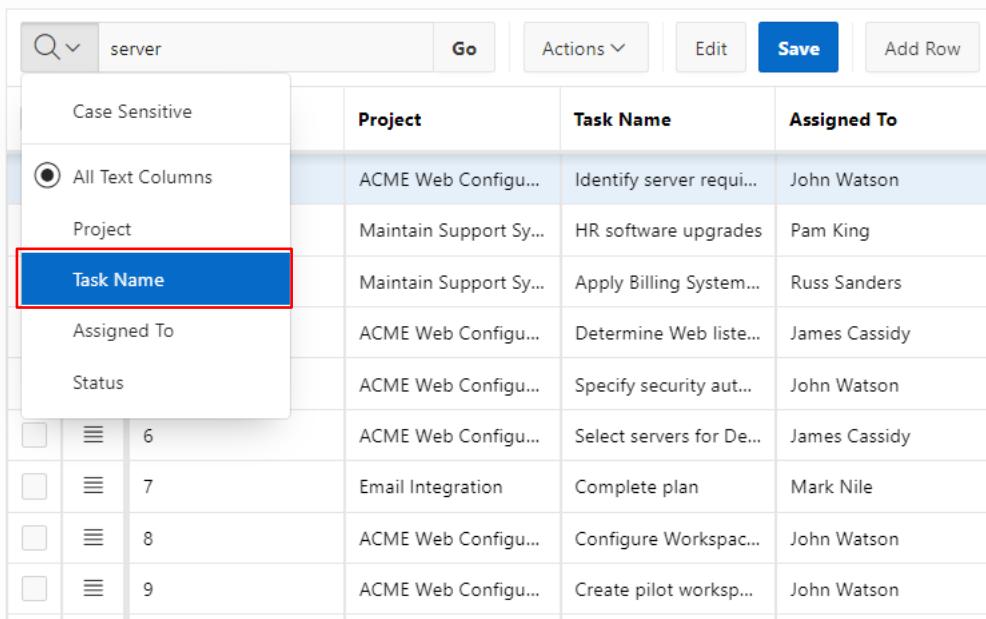
1. Notice that the interactive grid is now editable. You see the Edit, Save, and Add Row buttons. Also, the pagination type that you have set is displayed. Perform a non-case-sensitive search for **server** on the entire interactive grid.

To do this, enter **server** in the search bar text area and click **Go**.

| Project | Task Name | Assigned To | Start Date | Budget | End Date | Status | Cost |
|---------|-------------------------|----------------------------|---------------|------------|----------|------------|---------|
| 1 | ACME Web Configu... | Identify server requi... | John Watson | 11/19/2021 | 300 | 12/6/2021 | Closed |
| 6 | ACME Web Configu... | Select servers for De... | James Cassidy | 11/24/2021 | 600 | 11/26/2021 | Closed |
| 18 | Client Server Conver... | Identify pilot client s... | Scott Spencer | 12/28/2021 | 200 | 1/22/2022 | Closed |
| 19 | Public Website | Determine host server | Tiger Scott | 12/18/2021 | 200 | 12/19/2021 | Closed |
| 22 | Client Server Conver... | Migrate pilot client ... | Scott Spencer | 12/3/2021 | 500 | 12/22/2021 | Closed |
| 27 | Email Integration | Get RFPs for new se... | Mark Nile | 11/5/2021 | 1000 | 12/3/2021 | Open |
| 38 | Client Server Conver... | Post-migration review | Pam King | 1/17/2022 | 100 | 1/25/2022 | Closed |
| 39 | Client Server Conver... | Plan migration sche... | Pam King | 10/30/2021 | 100 | 11/25/2021 | Closed |
| 42 | Client Server Conver... | Migrate client server... | Pam King | 1/13/2022 | 12000 | 1/22/2022 | Open |
| 43 | Discussion Forum | Install web applicati... | Hank Davis | 12/14/2021 | 100 | 12/29/2021 | Closed |
| 50 | Migrate from Legac... | Obtain credentials | James Cassidy | 12/13/2021 | 500 | 12/26/2021 | Pending |
| 51 | Migrate from Legac... | Create new databas... | Scott Spencer | 10/30/2021 | 100 | 11/25/2021 | Pending |
| 52 | Migrate from Legac... | Identify integration ... | Mark Nile | 1/1/2022 | 2000 | 1/7/2022 | Pending |
| 56 | Email Integration | Purchase backup ser... | Mark Nile | 1/1/2022 | 3000 | 1/23/2022 | Pending |
| 58 | Migrate from Legac... | Map data usage | Mark Nile | 11/2/2021 | 8000 | 11/5/2021 | Pending |

- Remove the filter by clicking the X icon.

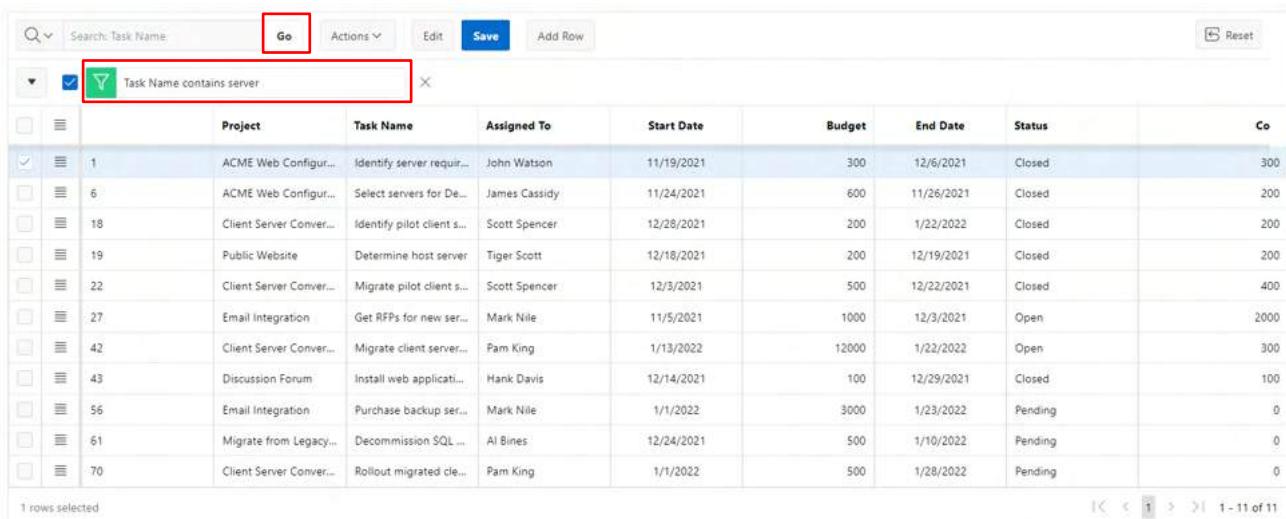
Now, in the search bar, click the magnifying glass and select the **Task Name** column.



The screenshot shows an interactive grid interface. At the top, there is a search bar with the text "server" and a "Go" button. Below the search bar are several filter buttons: "Case Sensitive", "All Text Columns" (which is selected), "Project", "Task Name" (which is highlighted with a red box), "Assigned To", and "Status". The main table has columns for "Project", "Task Name", and "Assigned To". The data rows include:

| Project | Task Name | Assigned To |
|------------------------|--------------------------|---------------|
| ACME Web Configu... | Identify server requi... | John Watson |
| Maintain Support Sy... | HR software upgrades | Pam King |
| Maintain Support Sy... | Apply Billing System... | Russ Sanders |
| ACME Web Configu... | Determine Web liste... | James Cassidy |
| ACME Web Configu... | Specify security aut... | John Watson |
| ACME Web Configu... | Select servers for De... | James Cassidy |
| Email Integration | Complete plan | Mark Nile |
| ACME Web Configu... | Configure Workspac... | John Watson |
| ACME Web Configu... | Create pilot worksp... | John Watson |

- Enter **server** in the text area and click **Go**. Notice that the search is now restricted to the **Task Name** column.

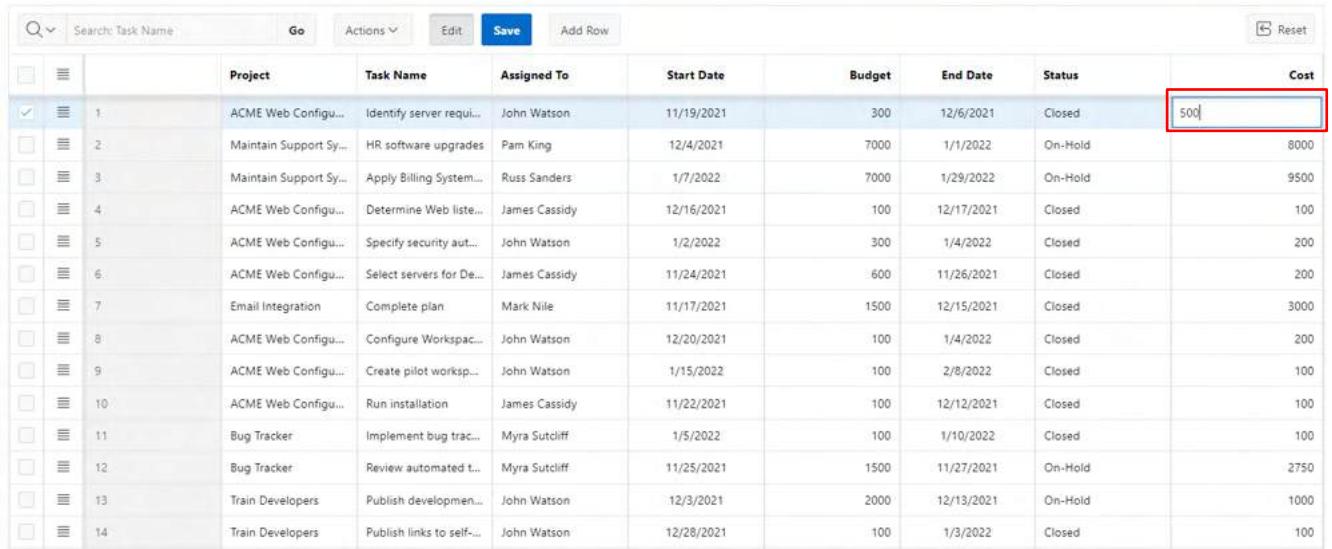


The screenshot shows an interactive grid interface with a search bar containing "Task Name contains server" and a magnifying glass icon. The "Go" button is highlighted with a red box. The main table has columns for "Project", "Task Name", "Assigned To", "Start Date", "Budget", "End Date", "Status", and "Co". The data rows include:

| Project | Task Name | Assigned To | Start Date | Budget | End Date | Status | Co |
|--------------------------|----------------------------|---------------|------------|--------|------------|---------|------|
| ACME Web Configur... | Identify server requi... | John Watson | 11/19/2021 | 300 | 12/6/2021 | Closed | 300 |
| ACME Web Configur... | Select servers for De... | James Cassidy | 11/24/2021 | 600 | 11/26/2021 | Closed | 200 |
| Client Server Convers... | Identify pilot client s... | Scott Spencer | 12/28/2021 | 200 | 1/22/2022 | Closed | 200 |
| Public Website | Determine host server | Tiger Scott | 12/18/2021 | 200 | 12/19/2021 | Closed | 200 |
| Client Server Convers... | Migrate pilot client s... | Scott Spencer | 12/3/2021 | 500 | 12/22/2021 | Closed | 400 |
| Email Integration | Get RFPs for new ser... | Mark Nile | 11/5/2021 | 1000 | 12/3/2021 | Open | 2000 |
| Client Server Convers... | Migrate client server... | Pam King | 1/13/2022 | 12000 | 1/22/2022 | Open | 300 |
| Discussion Forum | Install web applicati... | Hank Davis | 12/14/2021 | 100 | 12/29/2021 | Closed | 100 |
| Email Integration | Purchase backup ser... | Mark Nile | 1/1/2022 | 3000 | 1/23/2022 | Pending | 0 |
| Migrate from Legacy... | Decommission SQL ... | Al Bines | 12/24/2021 | 500 | 1/10/2022 | Pending | 0 |
| Client Server Convers... | Rollout migrated cle... | Pam King | 1/1/2022 | 500 | 1/28/2022 | Pending | 0 |

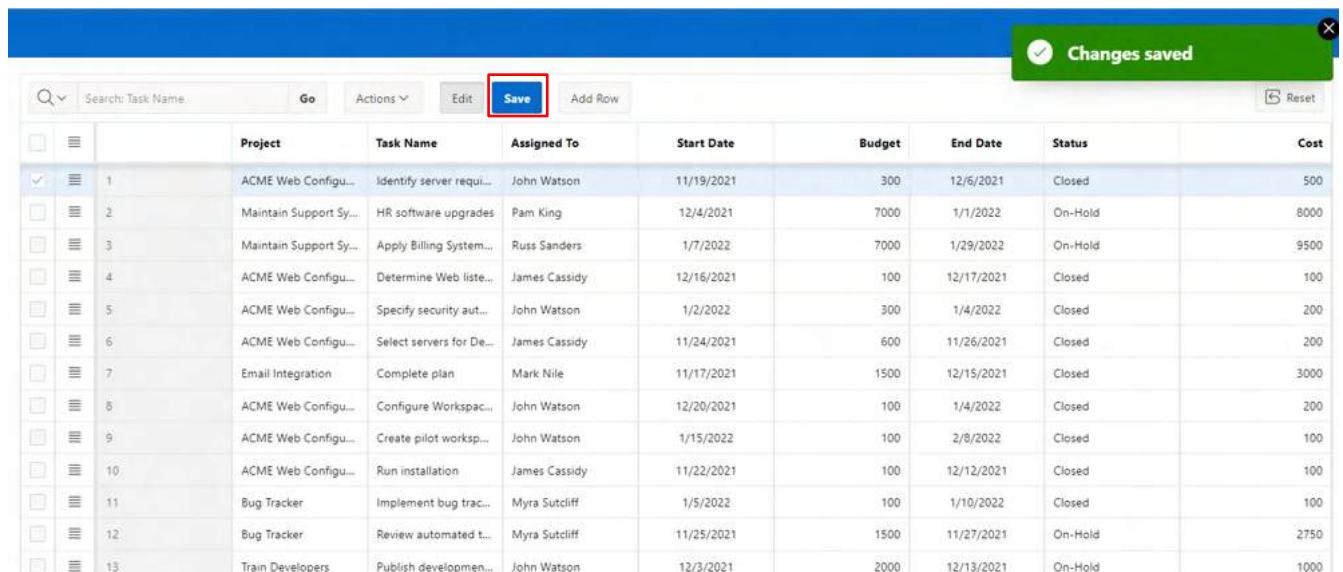
1 rows selected

4. Remove the filter by clicking the X icon. You want to update the Cost for the Project with ID
 1. Click the field and replace the existing value with **500**.



| | | Project | Task Name | Assigned To | Start Date | Budget | End Date | Status | Cost |
|-------------------------------------|----|------------------------|--------------------------|---------------|------------|--------|------------|---------|------|
| <input checked="" type="checkbox"/> | 1 | ACME Web Configu... | Identify server requi... | John Watson | 11/19/2021 | 300 | 12/6/2021 | Closed | 500 |
| <input type="checkbox"/> | 2 | Maintain Support Sy... | HR software upgrades | Pam King | 12/4/2021 | 7000 | 1/1/2022 | On-Hold | 8000 |
| <input type="checkbox"/> | 3 | Maintain Support Sy... | Apply Billing System... | Russ Sanders | 1/7/2022 | 7000 | 1/29/2022 | On-Hold | 9500 |
| <input type="checkbox"/> | 4 | ACME Web Configu... | Determine Web liste... | James Cassidy | 12/16/2021 | 100 | 12/17/2021 | Closed | 100 |
| <input type="checkbox"/> | 5 | ACME Web Configu... | Specify security aut... | John Watson | 1/2/2022 | 300 | 1/4/2022 | Closed | 200 |
| <input type="checkbox"/> | 6 | ACME Web Configu... | Select servers for De... | James Cassidy | 11/24/2021 | 600 | 11/26/2021 | Closed | 200 |
| <input type="checkbox"/> | 7 | Email Integration | Complete plan | Mark Nile | 11/17/2021 | 1500 | 12/15/2021 | Closed | 3000 |
| <input type="checkbox"/> | 8 | ACME Web Configu... | Configure Workspac... | John Watson | 12/20/2021 | 100 | 1/4/2022 | Closed | 200 |
| <input type="checkbox"/> | 9 | ACME Web Configu... | Create pilot worksp... | John Watson | 1/15/2022 | 100 | 2/8/2022 | Closed | 100 |
| <input type="checkbox"/> | 10 | ACME Web Configu... | Run installation | James Cassidy | 11/22/2021 | 100 | 12/12/2021 | Closed | 100 |
| <input type="checkbox"/> | 11 | Bug Tracker | Implement bug trac... | Myra Sutcliff | 1/5/2022 | 100 | 1/10/2022 | Closed | 100 |
| <input type="checkbox"/> | 12 | Bug Tracker | Review automated t... | Myra Sutcliff | 11/25/2021 | 1500 | 11/27/2021 | On-Hold | 2750 |
| <input type="checkbox"/> | 13 | Train Developers | Publish developmen... | John Watson | 12/3/2021 | 2000 | 12/13/2021 | On-Hold | 1000 |
| <input type="checkbox"/> | 14 | Train Developers | Publish links to self... | John Watson | 12/28/2021 | 100 | 1/3/2022 | Closed | 100 |

5. The changes are not saved yet. Click the **Save** button. The changes are now saved.



| | | Project | Task Name | Assigned To | Start Date | Budget | End Date | Status | Cost |
|-------------------------------------|----|------------------------|--------------------------|---------------|------------|--------|------------|---------|------|
| <input checked="" type="checkbox"/> | 1 | ACME Web Configu... | Identify server requi... | John Watson | 11/19/2021 | 300 | 12/6/2021 | Closed | 500 |
| <input type="checkbox"/> | 2 | Maintain Support Sy... | HR software upgrades | Pam King | 12/4/2021 | 7000 | 1/1/2022 | On-Hold | 8000 |
| <input type="checkbox"/> | 3 | Maintain Support Sy... | Apply Billing System... | Russ Sanders | 1/7/2022 | 7000 | 1/29/2022 | On-Hold | 9500 |
| <input type="checkbox"/> | 4 | ACME Web Configu... | Determine Web liste... | James Cassidy | 12/16/2021 | 100 | 12/17/2021 | Closed | 100 |
| <input type="checkbox"/> | 5 | ACME Web Configu... | Specify security aut... | John Watson | 1/2/2022 | 300 | 1/4/2022 | Closed | 200 |
| <input type="checkbox"/> | 6 | ACME Web Configu... | Select servers for De... | James Cassidy | 11/24/2021 | 600 | 11/26/2021 | Closed | 200 |
| <input type="checkbox"/> | 7 | Email Integration | Complete plan | Mark Nile | 11/17/2021 | 1500 | 12/15/2021 | Closed | 3000 |
| <input type="checkbox"/> | 8 | ACME Web Configu... | Configure Workspac... | John Watson | 12/20/2021 | 100 | 1/4/2022 | Closed | 200 |
| <input type="checkbox"/> | 9 | ACME Web Configu... | Create pilot worksp... | John Watson | 1/15/2022 | 100 | 2/8/2022 | Closed | 100 |
| <input type="checkbox"/> | 10 | ACME Web Configu... | Run installation | James Cassidy | 11/22/2021 | 100 | 12/12/2021 | Closed | 100 |
| <input type="checkbox"/> | 11 | Bug Tracker | Implement bug trac... | Myra Sutcliff | 1/5/2022 | 100 | 1/10/2022 | Closed | 100 |
| <input type="checkbox"/> | 12 | Bug Tracker | Review automated t... | Myra Sutcliff | 11/25/2021 | 1500 | 11/27/2021 | On-Hold | 2750 |
| <input type="checkbox"/> | 13 | Train Developers | Publish developmen... | John Watson | 12/3/2021 | 2000 | 12/13/2021 | On-Hold | 1000 |

6. You want to update another row. This time, click the Row Actions menu icon at the edge of the row for the project with ID **2** and select **Single Row View**.

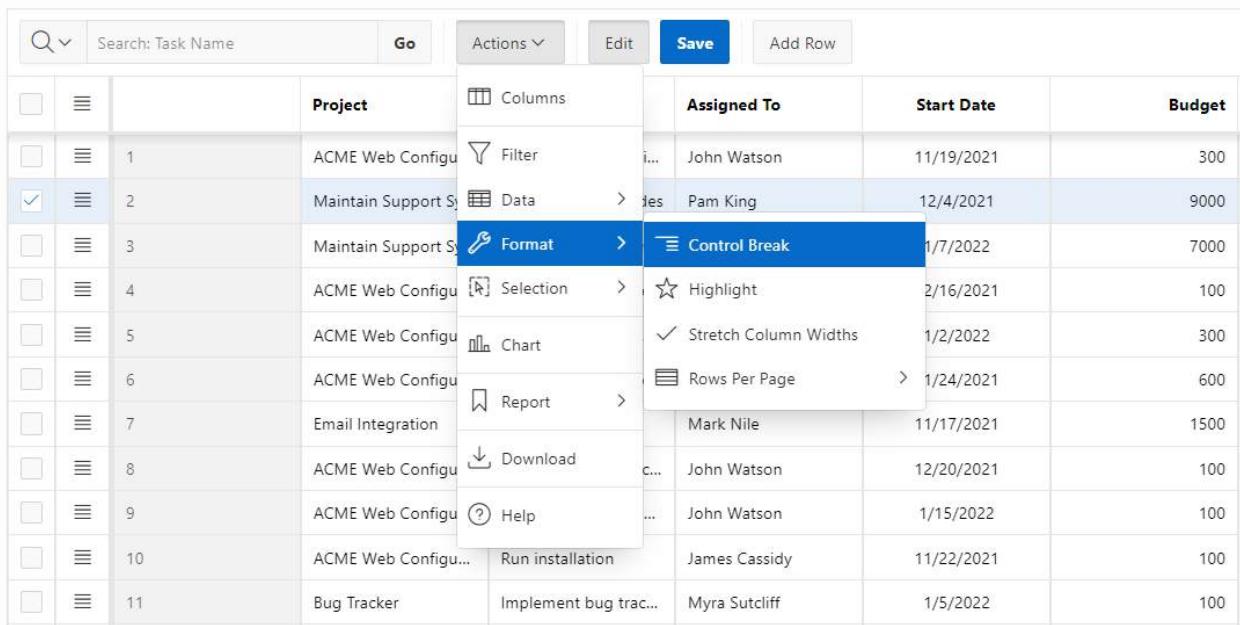
| | | | Project | Task Name | Assigned To |
|-------------------------------------|--------------------------|----|------------------------|--------------------------|---------------|
| | | 1 | ACME Web Configu... | Identify server requi... | John Watson |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | 2 | Maintain Support Sy... | HR software upgrades | Pam King |
| | | | Maintain Support Sy... | Apply Billing System... | Russ Sanders |
| | | | ACME Web Configu... | Determine Web liste... | James Cassidy |
| | | | ACME Web Configu... | Specify security aut... | John Watson |
| | | | ACME Web Configu... | Select servers for De... | James Cassidy |
| | | | Email Integration | Complete plan | Mark Nile |
| | | | ACME Web Configu... | Configure Workspac... | John Watson |
| | | | ACME Web Configu... | Create pilot worksp... | John Watson |
| | | 10 | ACME Web Configu... | Run installation | James Cassidy |
| | | 11 | Bug Tracker | Implement bug trac... | Myra Sutcliff |
| | | 12 | Bug Tracker | Review automated t... | Myra Sutcliff |

7. You are now in the single row view of the project with ID **2**. Replace the existing value for Budget with **9000** and click **Save**. Then, click **Report View**.

| Report View | | Actions | Reset |
|--------------|--------------------------|---------|-----------------|
| ID: | 2 | Changed | < Row 2 of 70 > |
| Project: | Maintain Support Systems | | |
| Task Name: | HR software upgrades | | |
| Assigned To: | Pam King | | |
| Start Date: | 12/4/2021 | | |
| Budget: | 9000 | | |
| End Date: | 1/1/2022 | | |
| Status: | On-Hold | | |
| Cost: | 8000 | | |

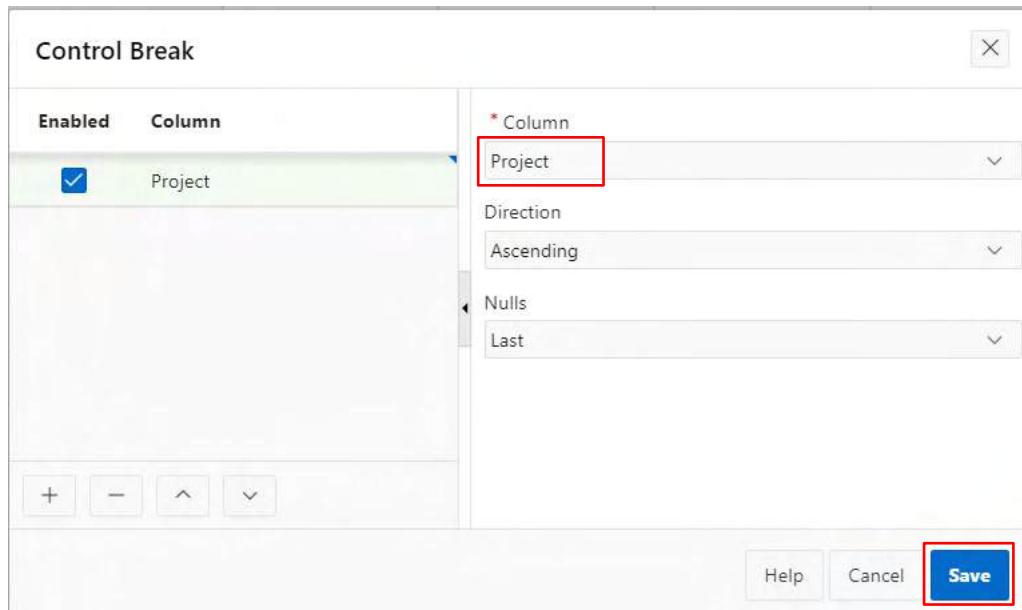
The row now displays **9000** for Budget.

8. You want to create a control break on the Project column. Click **Actions > Format > Control Break**.



| | | Project | Assigned To | Start Date | Budget |
|-------------------------------------|----|---------------------|---------------|------------|--------|
| | 1 | ACME Web Configu | John Watson | 11/19/2021 | 300 |
| <input checked="" type="checkbox"/> | 2 | Maintain Support S | Pam King | 12/4/2021 | 9000 |
| | 3 | Maintain Support S | | 1/7/2022 | 7000 |
| | 4 | ACME Web Configu | | 2/16/2021 | 100 |
| | 5 | ACME Web Configu | | 1/2/2022 | 300 |
| | 6 | ACME Web Configu | | 1/24/2021 | 600 |
| | 7 | Email Integration | Mark Nile | 11/17/2021 | 1500 |
| | 8 | ACME Web Configu | John Watson | 12/20/2021 | 100 |
| | 9 | ACME Web Configu | John Watson | 1/15/2022 | 100 |
| | 10 | ACME Web Configu... | James Cassidy | 11/22/2021 | 100 |
| | 11 | Bug Tracker | Myra Sutcliff | 1/5/2022 | 100 |

9. In the Control Break dialog box, for Column, enter **Project** and click **Save**.



| Enabled | Column |
|-------------------------------------|---------|
| <input checked="" type="checkbox"/> | Project |

* Column: Project

Direction: Ascending

Nulls: Last

Buttons: Help, Cancel, Save

10. The control break is now applied. You want to highlight rows that meet a condition. Select **Actions > Format > Highlight**.

The screenshot shows a grid interface with a toolbar at the top containing 'Search: Task Name', 'Go', 'Actions', 'Save', 'Edit', 'Add Row', and 'Reset'. A context menu is open from the 'Actions' button, with 'Format' selected. Under 'Format', 'Highlight' is highlighted with a blue background. The main grid area displays two sections: 'Project: ACME Web Configuration' and 'Project: Bug Tracker'. The 'ACME Web Configuration' section contains 10 rows of tasks, each with a checkbox in the first column. The 'Bug Tracker' section contains 6 rows of bugs. The columns are labeled 'Task Name', 'Start Date', 'Budget', 'End Date', 'Status', and 'Cost'.

| Task Name | Start Date | Budget | End Date | Status | Cost |
|-------------------------------|------------|--------|------------|---------|------|
| Identify server requirements | 12/20/2021 | 300 | 12/6/2021 | Closed | 500 |
| Determine Web architecture | | 100 | 12/17/2021 | Closed | 100 |
| Specify security protocols | | 300 | 1/4/2022 | Closed | 200 |
| Select servers for deployment | | 600 | 11/26/2021 | Closed | 200 |
| Configure Workstation | 12/20/2021 | 100 | 1/4/2022 | Closed | 200 |
| Create pilot workspace | 1/15/2022 | 100 | 2/8/2022 | Closed | 100 |
| Run installation | 11/22/2021 | 100 | 12/12/2021 | Closed | 100 |
| Implement bug tracking system | 1/5/2022 | 100 | 1/10/2022 | Closed | 100 |
| Review automated test cases | 11/25/2021 | 1500 | 11/27/2021 | On-Hold | 2750 |
| Document quality assurance | 11/4/2021 | 4000 | 11/11/2021 | Open | 3500 |
| Train developers on tools | 12/8/2021 | 2000 | 12/24/2021 | On-Hold | 0 |
| Measure effectiveness | 12/9/2021 | 1500 | 1/1/2022 | Pending | 0 |

11. In the Highlight dialog box, enter/select the following:

- Name: **Project Costing greater than 800**
- Background Color: **Yellow**
- Text Color: **Red**
- Column: **Cost**
- Operator: **greater than or equals**
- Value: **800**

Highlight

Name
Project Costing greater than 800

Highlight
Row

Background Color #ffff00 **Text Color** #ff0000 **Preview**

Condition Type
Column Cost **Operator** greater than or equals

Value 800

Save

12. Notice the rows with a cost greater than 800 are highlighted.

| Task Name | Assigned To | Start Date | Budget | End Date | Status | Cost |
|---------------------------------|---------------|------------|--------|------------|---------|------|
| Project: ACME Web Configuration | | | | | | |
| 1 Identify server require... | John Watson | 11/19/2021 | 300 | 12/6/2021 | Closed | 500 |
| 4 Determine Web listener... | James Cassidy | 12/16/2021 | 100 | 12/17/2021 | Closed | 100 |
| 5 Specify security authen... | John Watson | 1/2/2022 | 300 | 1/4/2022 | Closed | 200 |
| 6 Select servers for Devel... | James Cassidy | 11/24/2021 | 600 | 11/26/2021 | Closed | 200 |
| 8 Configure Workspace p... | John Watson | 12/20/2021 | 100 | 1/4/2022 | Closed | 200 |
| 9 Create pilot workspace | John Watson | 1/15/2022 | 100 | 2/8/2022 | Closed | 100 |
| 10 Run installation | James Cassidy | 11/22/2021 | 100 | 12/12/2021 | Closed | 100 |
| Project: Bug Tracker | | | | | | |
| 11 Implement bug trackin... | Myra Sutcliff | 1/5/2022 | 100 | 1/10/2022 | Closed | 100 |
| 12 Review automated testi... | Myra Sutcliff | 11/25/2021 | 1500 | 11/27/2021 | On-Hold | 2750 |
| 38 Document quality assur... | Myra Sutcliff | 11/4/2021 | 4000 | 11/11/2021 | Open | 3500 |
| 59 Train developers on tra... | Myra Sutcliff | 12/9/2021 | 2000 | 12/24/2021 | On-Hold | 0 |
| 60 Measure effectiveness ... | Myra Sutcliff | 12/9/2021 | 1500 | 1/1/2022 | Pending | 0 |

13. You want to save the changes made to the interactive grid. Select **Actions > Report > Save As**.

The screenshot shows an Oracle Interactive Grid interface. At the top, there's a search bar labeled 'Search: Task Name' and a 'Go' button. To the right are 'Actions', 'Edit', and a blue 'Save' button. Below the search bar is a toolbar with icons for 'Project', 'Columns', 'Filter', 'Data', 'Format', 'Selection', and 'Chart'. The 'Chart' option is currently selected and highlighted in blue. A context menu is open, listing options like 'Report', 'Save', 'Download', 'Edit', 'Delete', and 'Reset'. The 'Save' option is also highlighted with a red box. The main grid area displays a table with columns 'Task Name', 'Start Date', 'Budget', and 'End Date'. One row is highlighted in yellow and contains a star icon. The data in the grid includes tasks like 'Identify server requirements', 'Determine Web', etc., with dates ranging from 11/19/2021 to 1/5/2022.

14. In the Report – Save As dialog box, for Type, select **Private**. Under Name, enter **My Private Report**. Click **Save**.

The screenshot shows the 'Report - Save As' dialog box. It has a header 'Report - Save As' and a close button. The 'Type' section shows 'Private' selected. The 'Name' section has a required field indicator '*' and a text input field containing 'My Private Report', which is also highlighted with a red box. At the bottom right are buttons for 'Help', 'Cancel', and a large blue 'Save' button, which is also highlighted with a red box.

15. Notice that the Primary interactive grid and the interactive grid you saved are now available in the Reports drop-down list.

You want to return to the Primary interactive grid. Click **Primary Report** in the Reports drop-down list.

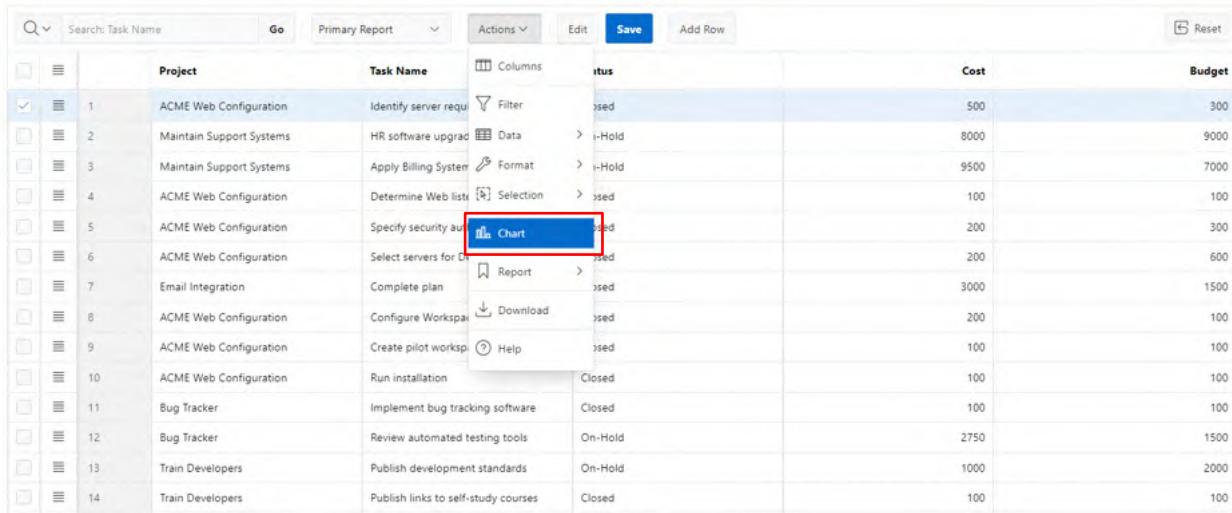
The screenshot shows the Oracle Database Interactive Grid interface. At the top, there is a search bar labeled "Search: Task Name" with a "Go" button. To the right of the search bar is a dropdown menu titled "My Private Report" with a "Default" option. Below the search bar is a toolbar with "Actions", "Edit", "Save", and "Add Row" buttons. A yellow banner at the top right says "Project Costing grea...". The main area is a table with columns: "Task Name", "Assigned To", "Start Date", and "Budget". The table has two sections: "Project: ACME Web Configuration" and "Project: Bug Tracker". The first section contains 10 rows of task data. The second section contains 1 row of task data. The "Start Date" column header for the first section is currently being edited, indicated by a small edit icon in the header cell.

16. You want to make few more customizations and save the interactive grid as another Private report. You do not want the **Start Date**, **End Date**, and **Assigned To** columns to be displayed in the report. Click the **Start Date** column header and then click **Hide**.

The screenshot shows the Oracle Database Interactive Grid interface after performing the customization. The "Start Date" column header is now hidden, as indicated by the "Hide" option in the context menu. The table structure remains the same, with the "Start Date" column header replaced by a standard header cell. The other columns are visible: "Project", "Task Name", "End Date", "Status", "Assigned To", "Cost", and "Budget". The data in the table is identical to the previous screenshot.

Similarly, perform the same steps for the **End Date** and **Assigned To** columns.

17. You want to add a chart to the interactive grid. Select **Actions > Chart**.

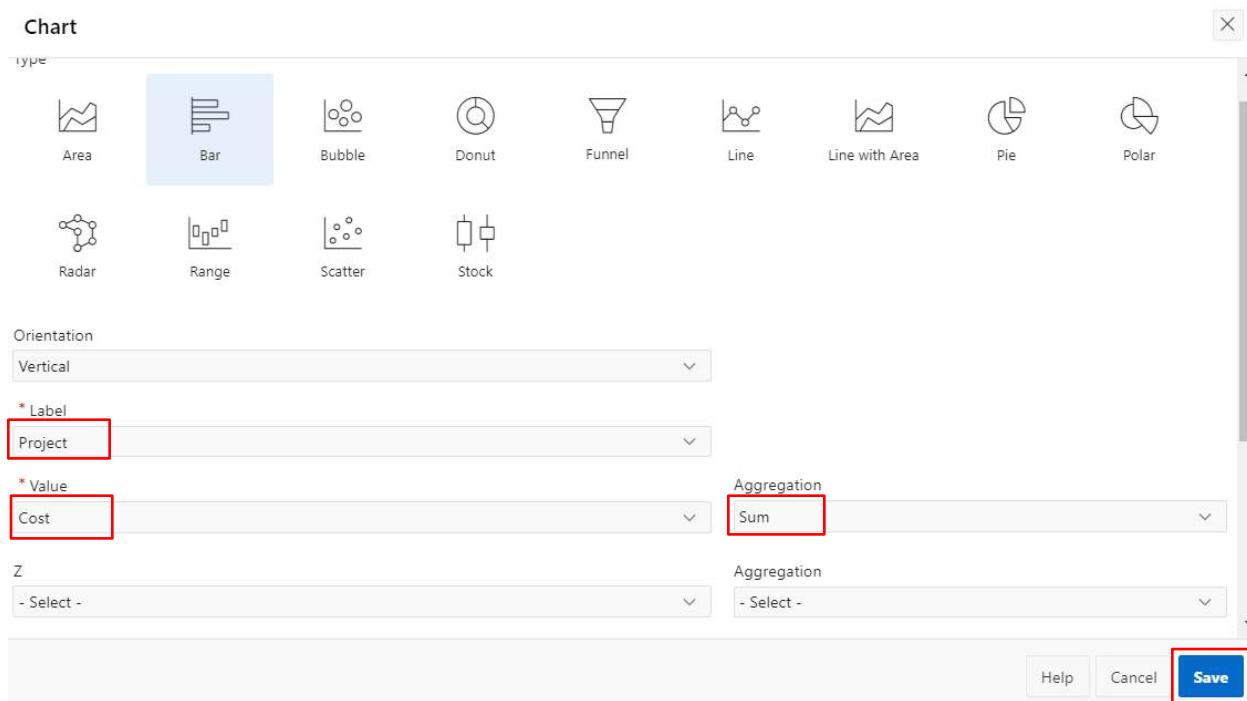


| Project | Task Name | Status | Cost | Budget |
|----------------------------|-------------------------------------|---------|------|--------|
| 1 ACME Web Configuration | Identify server requirements | On-Hold | 500 | 300 |
| 2 Maintain Support Systems | HR software upgrade | On-Hold | 8000 | 9000 |
| 3 Maintain Support Systems | Apply Billing System | On-Hold | 9500 | 7000 |
| 4 ACME Web Configuration | Determine Web lists | On-Hold | 100 | 100 |
| 5 ACME Web Configuration | Specify security audit | On-Hold | 200 | 300 |
| 6 ACME Web Configuration | Select servers for Dev | On-Hold | 200 | 600 |
| 7 Email Integration | Complete plan | On-Hold | 3000 | 1500 |
| 8 ACME Web Configuration | Configure Workspace | On-Hold | 200 | 100 |
| 9 ACME Web Configuration | Create pilot workspace | On-Hold | 100 | 100 |
| 10 ACME Web Configuration | Run installation | Closed | 100 | 100 |
| 11 Bug Tracker | Implement bug tracking software | Closed | 100 | 100 |
| 12 Bug Tracker | Review automated testing tools | On-Hold | 2750 | 1500 |
| 13 Train Developers | Publish development standards | On-Hold | 1000 | 2000 |
| 14 Train Developers | Publish links to self-study courses | Closed | 100 | 100 |

18. In the Chart dialog box, select:

- Type: **Bar**
- Label: **Project**
- Value: **Cost**
- Aggregation: **Sum**

Click **Save**.



Chart

Type

Bar

Orientation

Vertical

* Label

Project

* Value

Cost

Aggregation

Sum

Z

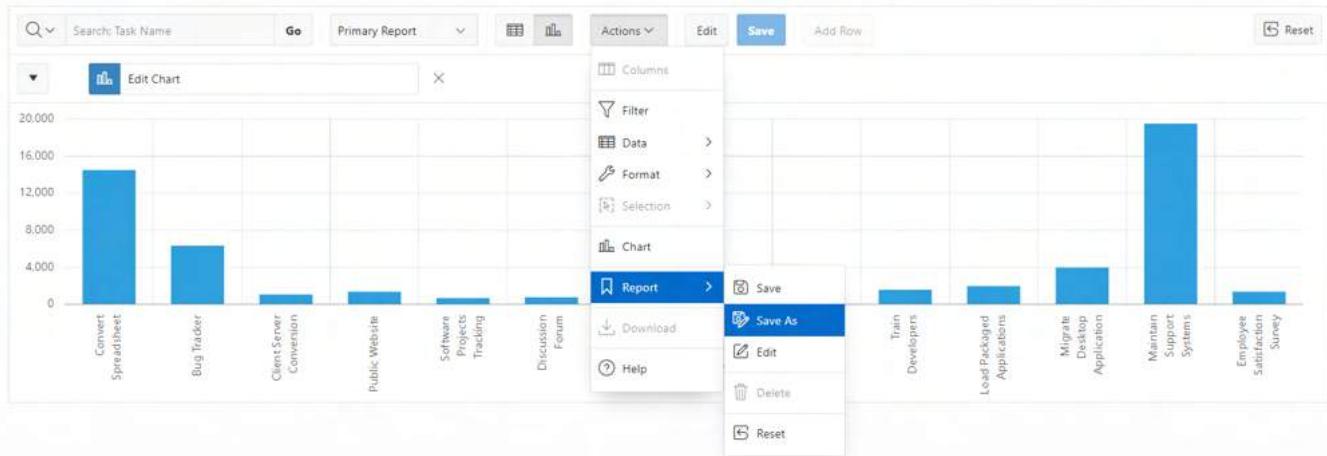
- Select -

Aggregation

- Select -

Help Cancel Save

19. The chart is displayed. You want to save the customization made to the interactive grid. Select **Actions > Report > Save As**.



20. In the Report – Save As dialog box, for Type, select **Private**. Under Name, enter **My Custom Report**. Then, click **Save**.

The 'Report - Save As' dialog box is shown. The 'Type' field contains 'Private'. The 'Name' field contains 'My Custom Report', which is highlighted with a red box. At the bottom right of the dialog are 'Help', 'Cancel', and 'Save' buttons.

21. The report is now saved under Private in the Reports drop-down list. Click the **Grid** icon.



22. You want to download the report. Select **Actions > Download**.

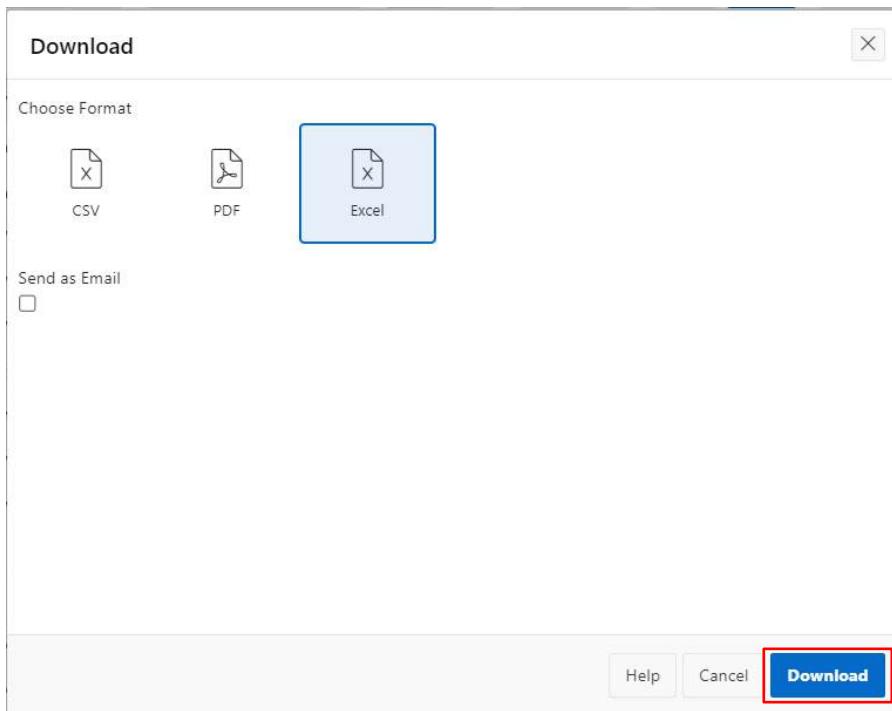
The screenshot shows an interactive grid with a context menu open over the 1st row. The menu options are:

- Columns
- Filter
- Data
- Format
- Selection
- Chart
- Report
- Download** (highlighted in blue)
- Help
- Closed

The grid displays a list of tasks across various projects. The columns are labeled: Project, Task Name, Cost, and Budget. The data is as follows:

| Project | Task Name | Cost | Budget |
|----------------------------|---|------|--------|
| 1 ACME Web Configuration | Identify server requirements | 500 | 300 |
| 2 Maintain Support Systems | HR software upgrades | 8000 | 9000 |
| 3 Maintain Support Systems | Apply Billing System updates | 9500 | 7000 |
| 4 ACME Web Configuration | Determine Web listener configuration | 100 | 100 |
| 5 ACME Web Configuration | Specify security authentication schema | 200 | 300 |
| 6 ACME Web Configuration | Select servers for Development Test Environment | 200 | 600 |
| 7 Email Integration | Complete plan | 3000 | 1500 |
| 8 ACME Web Configuration | Configure Workspace provisioning | 200 | 100 |
| 9 ACME Web Configuration | Create pilot workspace | 100 | 100 |
| 10 ACME Web Configuration | Run installation | 100 | 100 |
| 11 Bug Tracker | Implement bug tracking software | 100 | 100 |
| 12 Bug Tracker | Review automated testing tools | 2750 | 1500 |
| 13 Train Developers | Publish development standards | 1000 | 2000 |
| 14 Train Developers | Publish links to self-study courses | 100 | 100 |
| 15 Train Developers | Create training workspace | 500 | 700 |

23. Note that the **HTML** download option is no longer available. Select **Excel** and click **Download**.



24. The report is now downloaded as an **Excel file**.

| A | B | C | D | E | F | G |
|-------------------------------|--|---------|------|------|---|---|
| Project | Task Name | Status | | | | |
| 2 ACME Web Configuration | Identify server requirements | Closed | 500 | 300 | | |
| 3 Maintain Support Systems | HR software upgrades | On-Hold | 8000 | 9000 | | |
| 4 Maintain Support Systems | Apply Billing System updates | On-Hold | 9500 | 7000 | | |
| 5 ACME Web Configuration | Determine Web listener configuration(s) | Closed | 100 | 100 | | |
| 6 ACME Web Configuration | Specify security authentication schemes | Closed | 200 | 300 | | |
| 7 ACME Web Configuration | Select servers for Development, Test, Production | Closed | 200 | 600 | | |
| 8 Email Integration | Complete plan | Closed | 3000 | 1500 | | |
| 9 ACME Web Configuration | Configure Workspace provisioning | Closed | 200 | 100 | | |
| 10 ACME Web Configuration | Create pilot workspace | Closed | 100 | 100 | | |
| 11 ACME Web Configuration | Run installation | Closed | 100 | 100 | | |
| 12 Bug Tracker | Implement bug tracking software | Closed | 100 | 100 | | |
| 13 Bug Tracker | Review automated testing tools | On-Hold | 2750 | 1500 | | |
| 14 Train Developers | Publish development standards | On-Hold | 1000 | 2000 | | |
| 15 Train Developers | Publish links to self-study courses | Closed | 100 | 100 | | |
| 16 Train Developers | Create training workspace | Closed | 500 | 700 | | |
| 17 Load Packaged Applications | Identify point solutions required | Closed | 200 | 300 | | |
| 18 Software Projects Tracking | Conduct project kickoff meeting | Closed | 100 | 100 | | |
| 19 Client Server Conversion | Identify pilot client server applications | Closed | 200 | 200 | | |
| 20 Public Website | Determine host server | Closed | 200 | 200 | | |
| 21 Load Packaged Applications | Install in development | Closed | 100 | 100 | | |
| 22 Public Website | Check software licenses | Closed | 100 | 100 | | |

You now know how to manage and customize an interactive report as an end user. You may now **proceed to the next practice**.

Practice: Creating Application Page Controls

Practice 1: Add Items and Buttons to a Page

Overview

In this practice, you create new page items and buttons on the Shopping Cart and Add to Cart pages created in **Practice 2** of the **Developing Reports** workshop.

Customers will be able to:

- Review items in the shopping cart
- Edit the quantity of the items
- Remove an item
- Clear the shopping cart
- Proceed to checkout

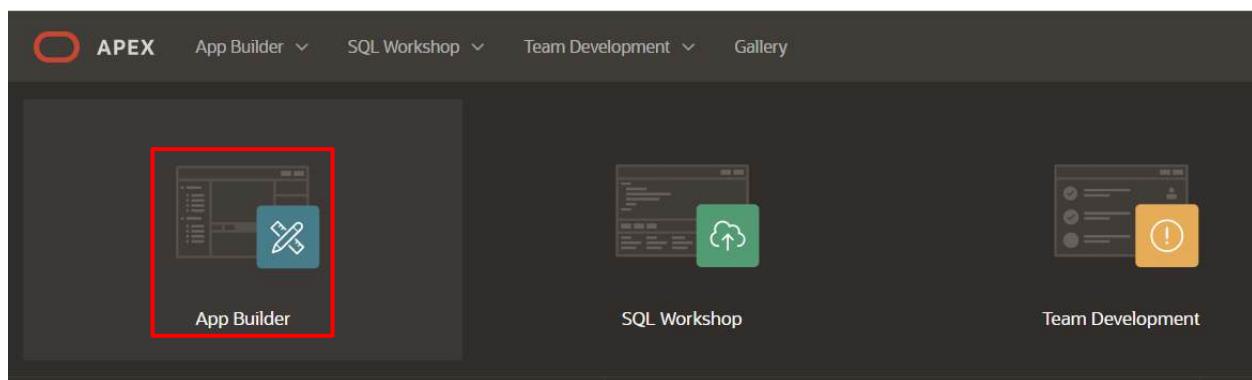
Downloads

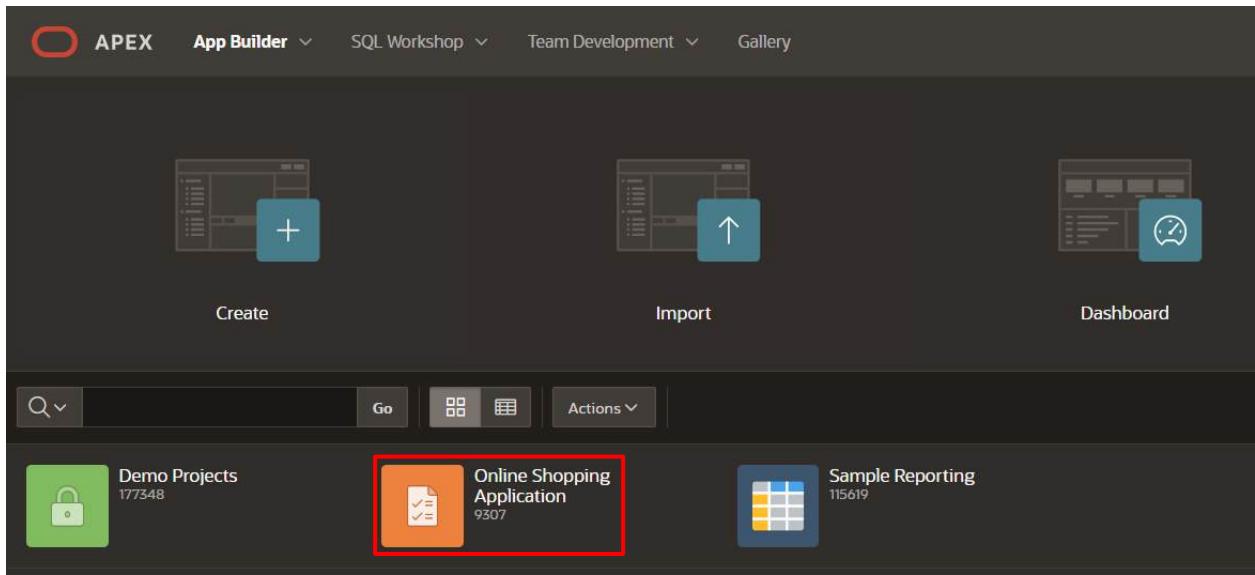
- Did you miss out on trying the previous practices? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, follow the steps described in the [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Tasks

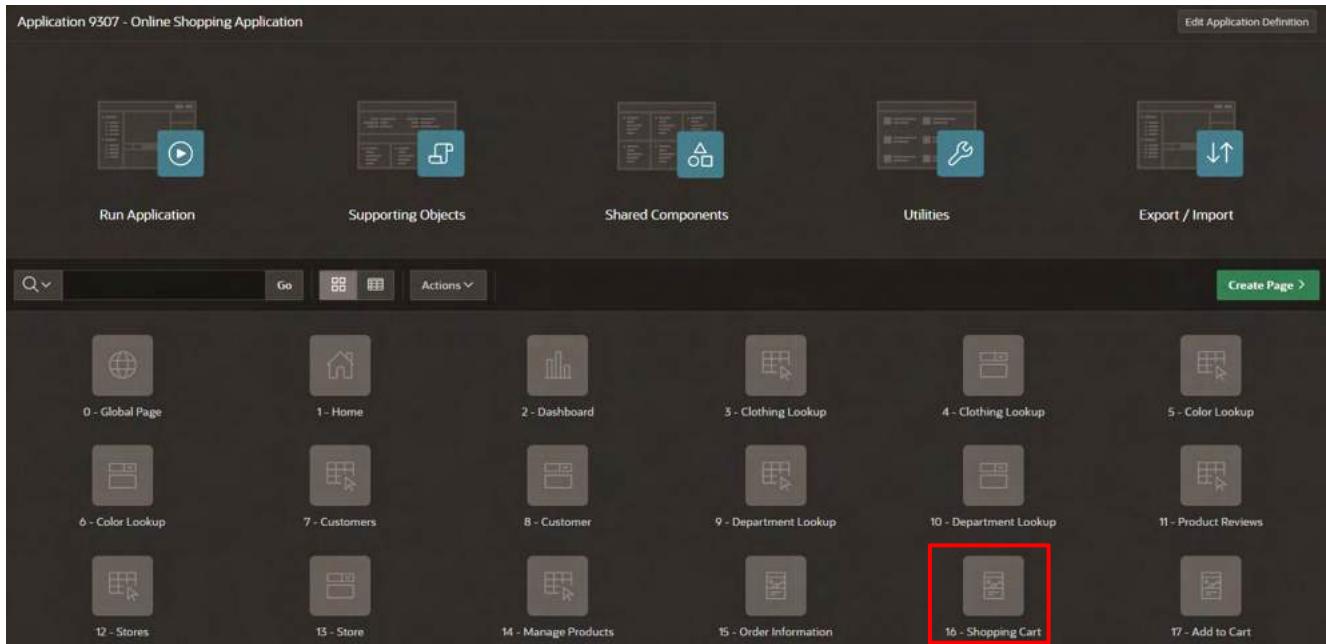
Add Items and Buttons to a Page

1. Navigate to the **App Builder**. Then click **Online Shopping Application**.

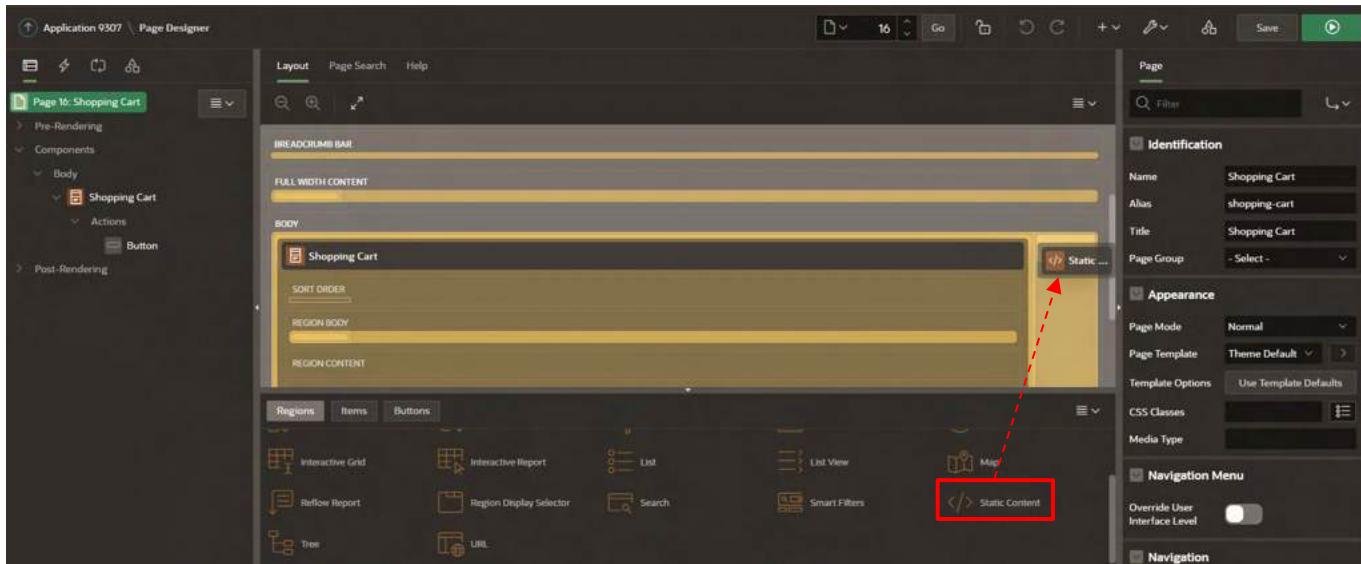




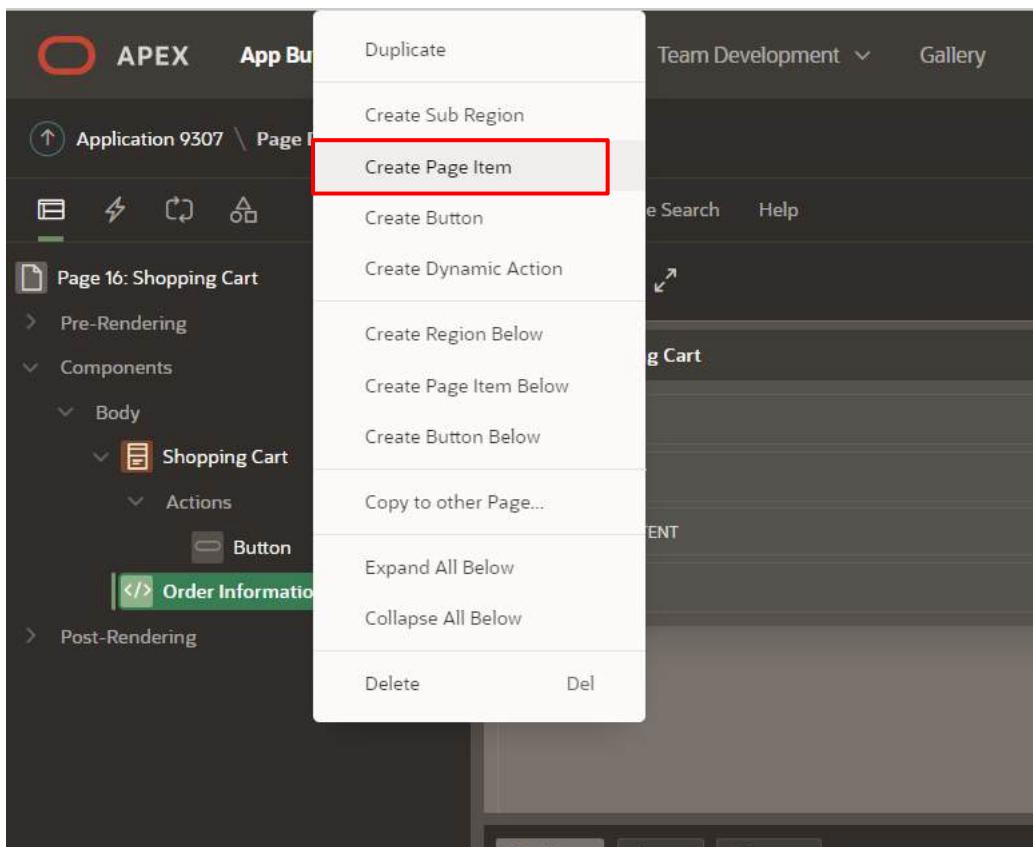
2. Under **Page Icons**, select **Shopping Cart**.



3. Drag a **Static Content** region and drop it to the right of the Shopping Cart region to create a second region of content.

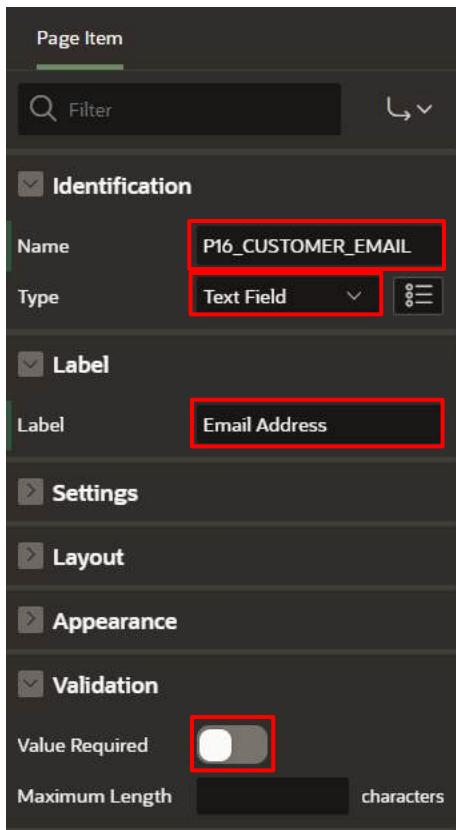


4. In the Property Editor, for Title, enter **Order Information**.
5. Navigate to the **Order Information** (left pane) region.
6. Right-click the **Order Information** region and select **Create Page Item**.



7. In the Property Editor, enter/select the following:

- Name: **P16_CUSTOMER_EMAIL**
- Type: **Text Field**
- Label: **Email Address**
- Value Required: **Off**



8. Create four items as follows:

Table 1: Create Application Page Controls | Practice 1: Add Items and Buttons to a Page

| Name | Type | Label | Template | Value Required |
|-----------------------|------------|-----------|---------------------|----------------|
| P16_CUSTOMER_FULLNAME | Text Field | Full Name | Optional - Floating | Off |
| P16_ORDER_ID | Hidden | | | |
| P16_CUSTOMER_ID | Hidden | | | |

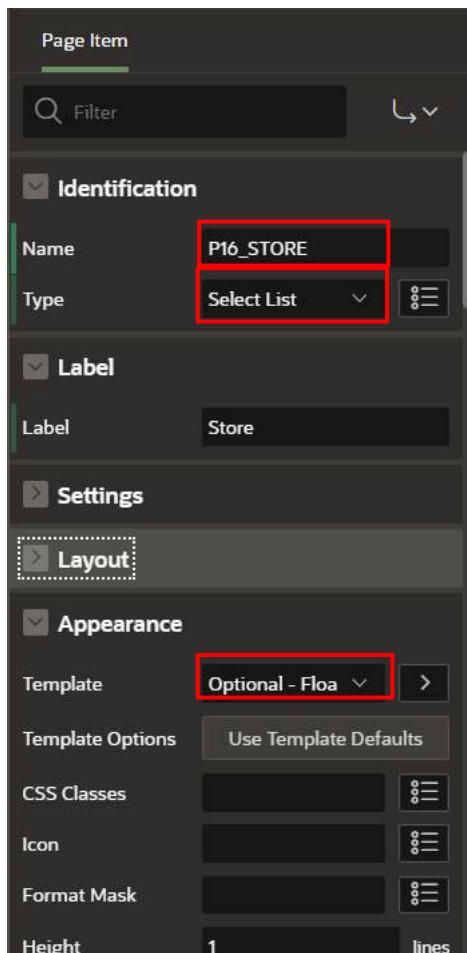
| | | | | |
|-----------|-------------|-------|---------------------|-----|
| P16_STORE | Select List | Store | Optional - Floating | Off |
|-----------|-------------|-------|---------------------|-----|

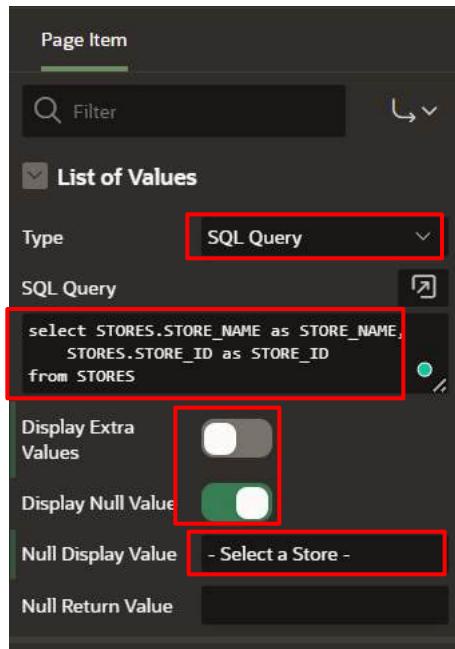
For the **P16_STORE** item, in the list of values section, configure the type as follows:

- Type: Select **SQL Query**
- SQL Query: Enter the following SQL Query:

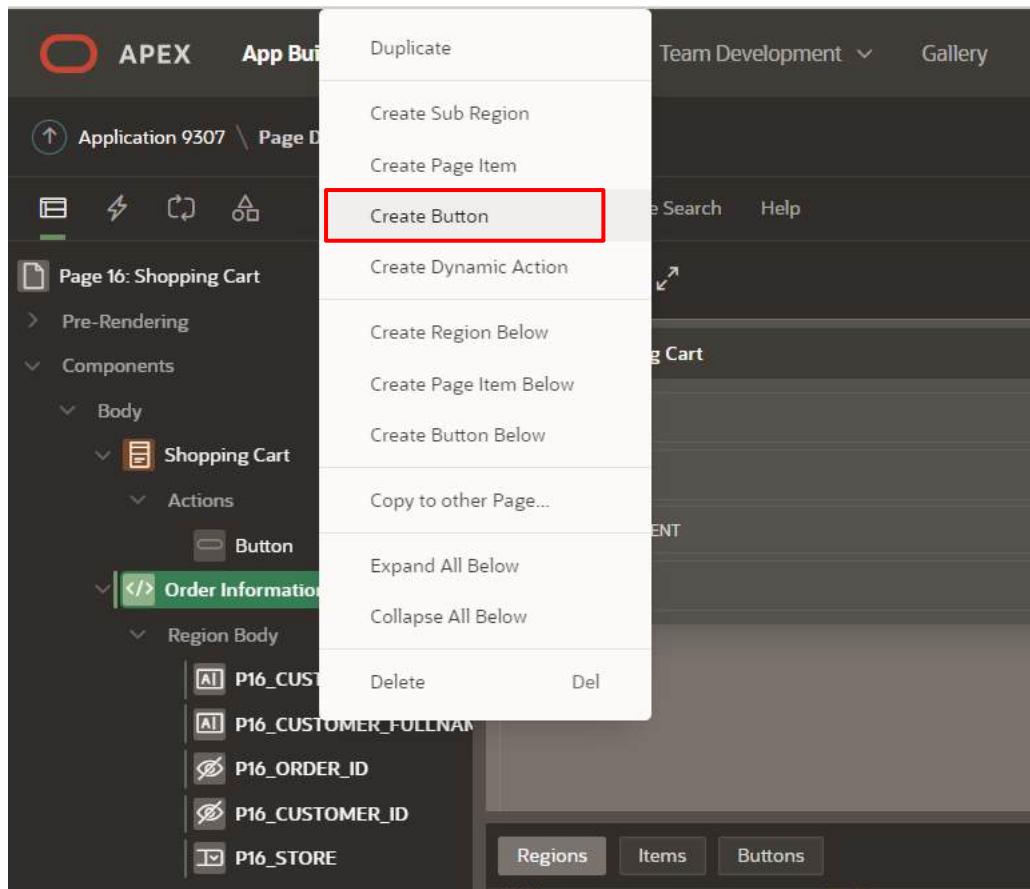
```
select STORES.STORE_NAME as STORE_NAME,
       STORES.STORE_ID as STORE_ID
  from STORES
```

- Set Display Extra Values to **Off**
- Null Display Value: Enter - **Select a Store -**





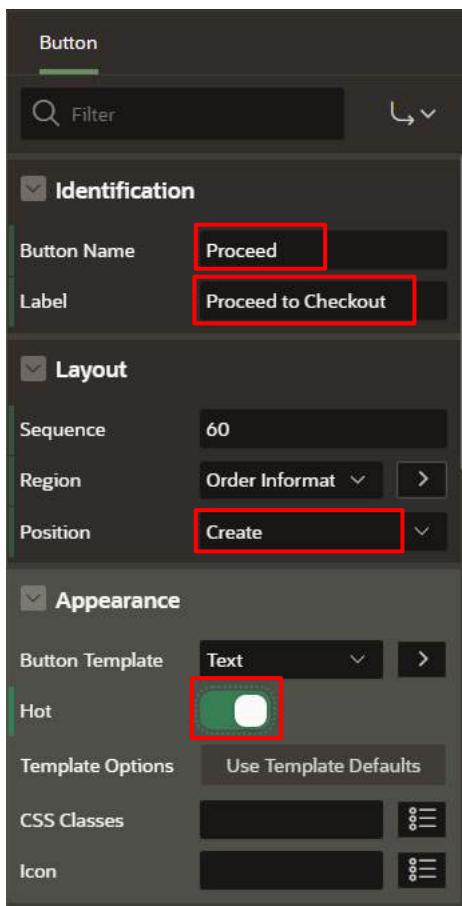
9. Navigate to the **Order Information** (left pane) region.
10. Right-click the **Order Information** region and select **Create Button**.

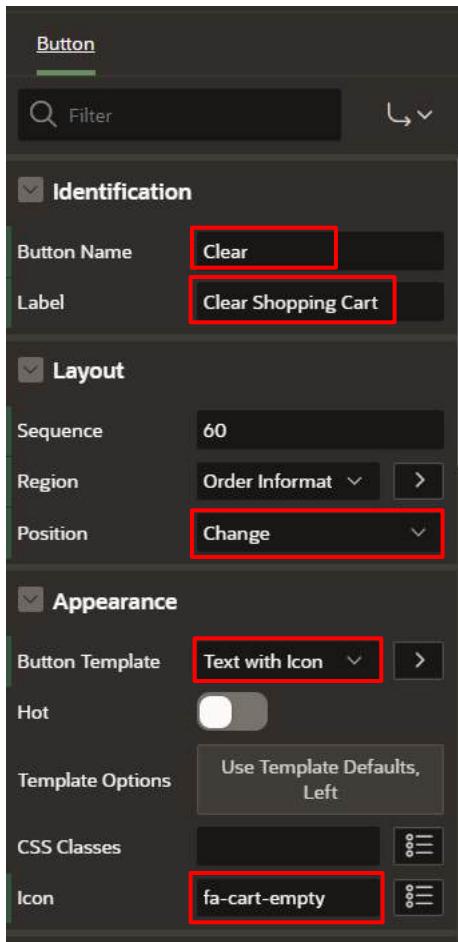


11. Create two buttons as follows:

Table 2: Create Application Page Controls | Practice 1: Add Items and Buttons to a Page

| Button Name | Label | Button Position | Button Template | Hot | Icon |
|-------------|---------------------|-----------------|-----------------|-----|---------------|
| Proceed | Proceed to Checkout | Create | Text | On | |
| Clear | Clear Shopping Cart | Change | Text with Icon | Off | fa-cart-empty |

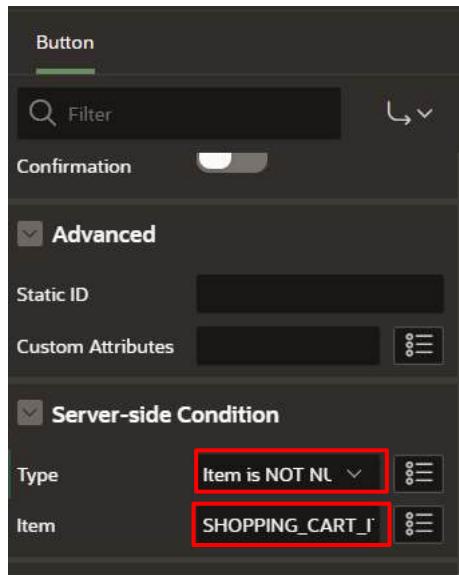




Under Server-side Condition:

Table 3: Create Application Page Controls | Practice 1: Add Items and Buttons to a Page

| Button Name | Type | Item |
|-------------|------------------|---------------------|
| Proceed | Item is NOT NULL | SHOPPING_CART_ITEMS |
| Clear | Item is NOT NULL | SHOPPING_CART_ITEMS |



Add Items and Buttons

In this task, you will create four page items:

- PRODUCT_ID: To get the product ID
- ACTION: To identify the action (Add/Edit/Delete) made for the customer
- QUANTITY: To permit customers to select the number of items to add or edit in the shopping cart
- SHOPPING_CART_ITEMS: To get the total number of items in the shopping cart after an action is made

1. Navigate to **Page Finder** and click **File**. In the **Page Finder** pop-up, select **Page 17**.

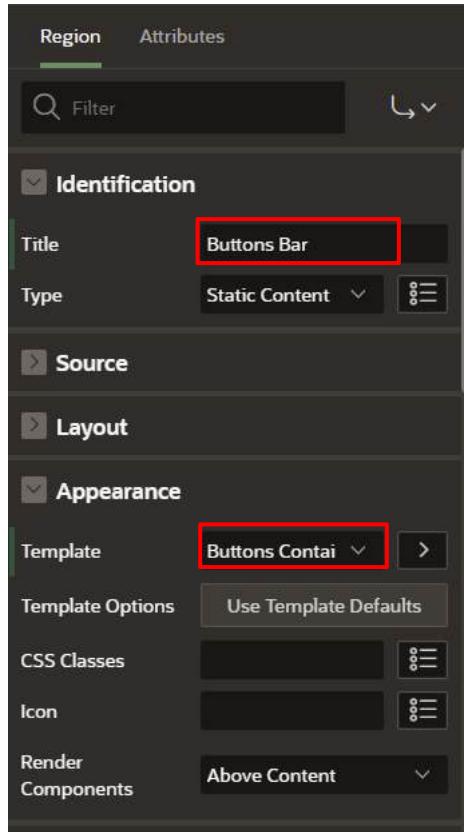
The screenshot shows the Oracle ADF Page Designer's Page Finder dialog. The 'All Pages' tab is selected. The list includes various pages like Product Reviews, Stores, and Add to Cart. The 'Add to Cart' page (ID 17) is highlighted with a red box. The top right corner of the dialog also has a red box around it.

2. Drag a **Static Content** region and drop it into the **Dialog Footer**.

The screenshot shows the Oracle ADF Page Designer interface for 'Page 17: Add to Cart'. The left sidebar shows components like Dialog Header, Content Body, and Dialog Footer. The main area shows the page structure with sections like REGION BODY, REGION CONTENT, and DIALOG FOOTER. A 'Static Content' region is selected and highlighted with a red box. A dashed red arrow points from this region towards the DIALOG FOOTER section, which is also highlighted with a red box. The right side of the screen displays the page's properties panel.

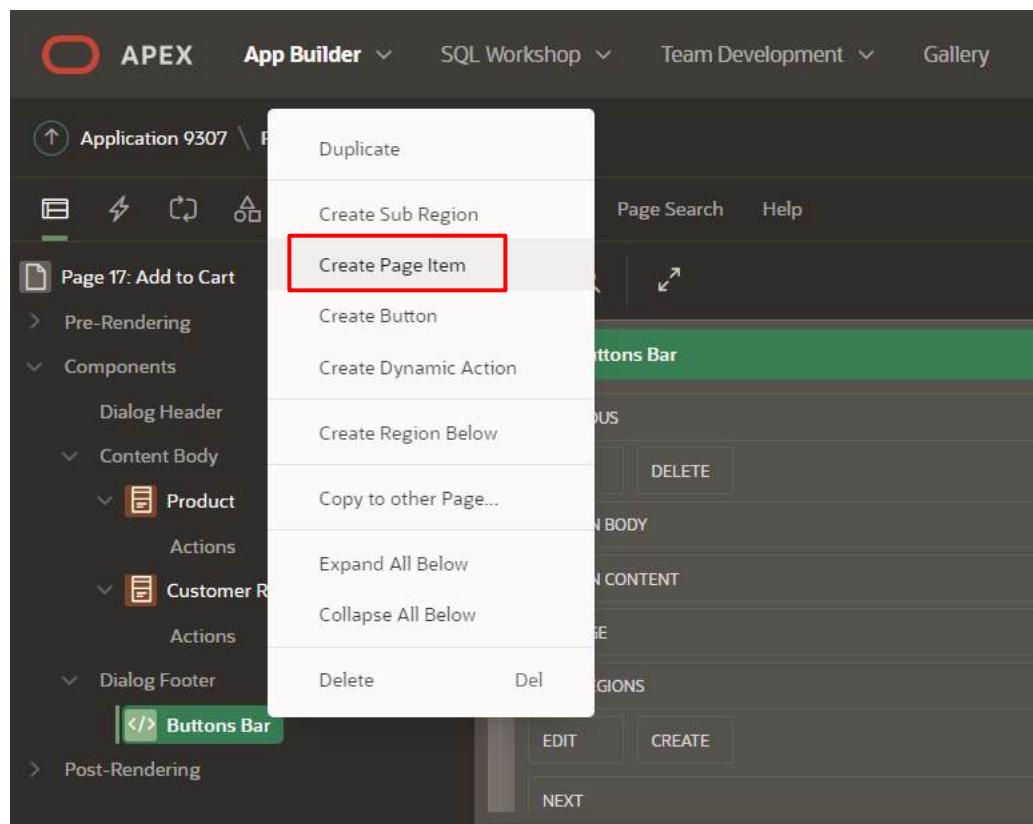
3. In the Property Editor, enter/select the following:

- Title: **Buttons Bar**
- Template: **Buttons Container**



4. In the Rendering tree (left pane), navigate to the **Buttons Bar** region.

5. Right-click the **Buttons Bar** region and select **Create Page Item**.



6. Create four items in the Property Editor:

Table 4: Create Application Page Controls | Practice 1: Add Items and Buttons to a Page

| Name | Type | Label | Template |
|-------------------------|-------------|----------|----------|
| P17_ACTION | Hidden | | |
| P17_PRODUCT_ID | Hidden | | |
| P17_SHOPPING_CART_ITEMS | Hidden | | |
| P17_QUANTITY | Select List | Quantity | Required |

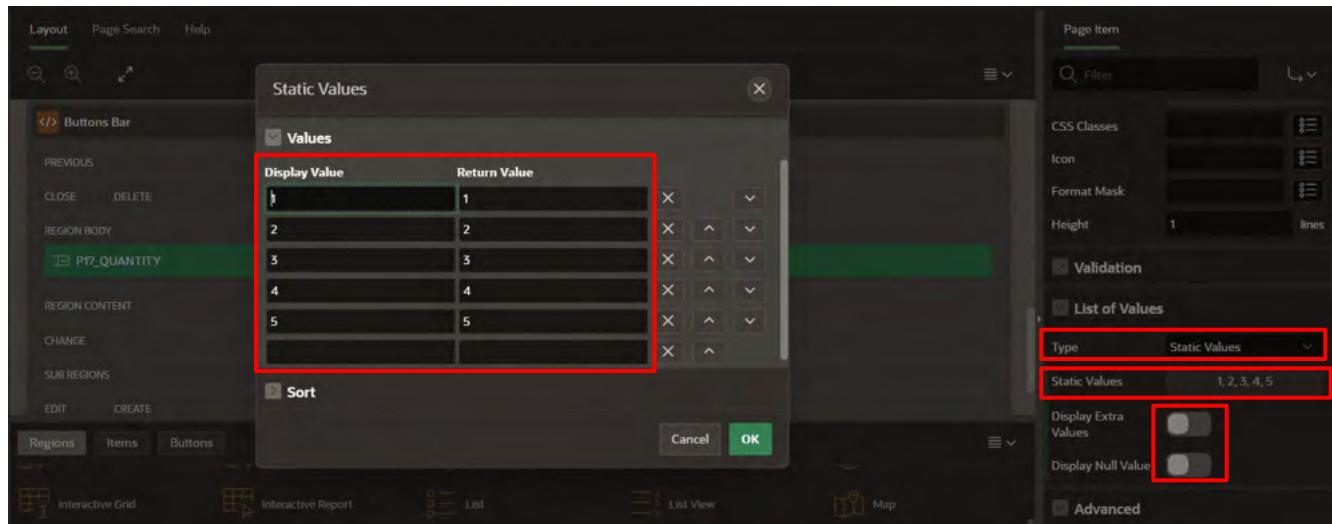
For the **P17_QUANTITY** item, do the following:

- Under the List of Values section:
 - a) For Type, select **Static Values**.
 - b) For Static Values , click **Display1**, **Display2** and enter the following:

Table 5: Create Application Page Controls | Practice 1: Add Items and Buttons to a Page

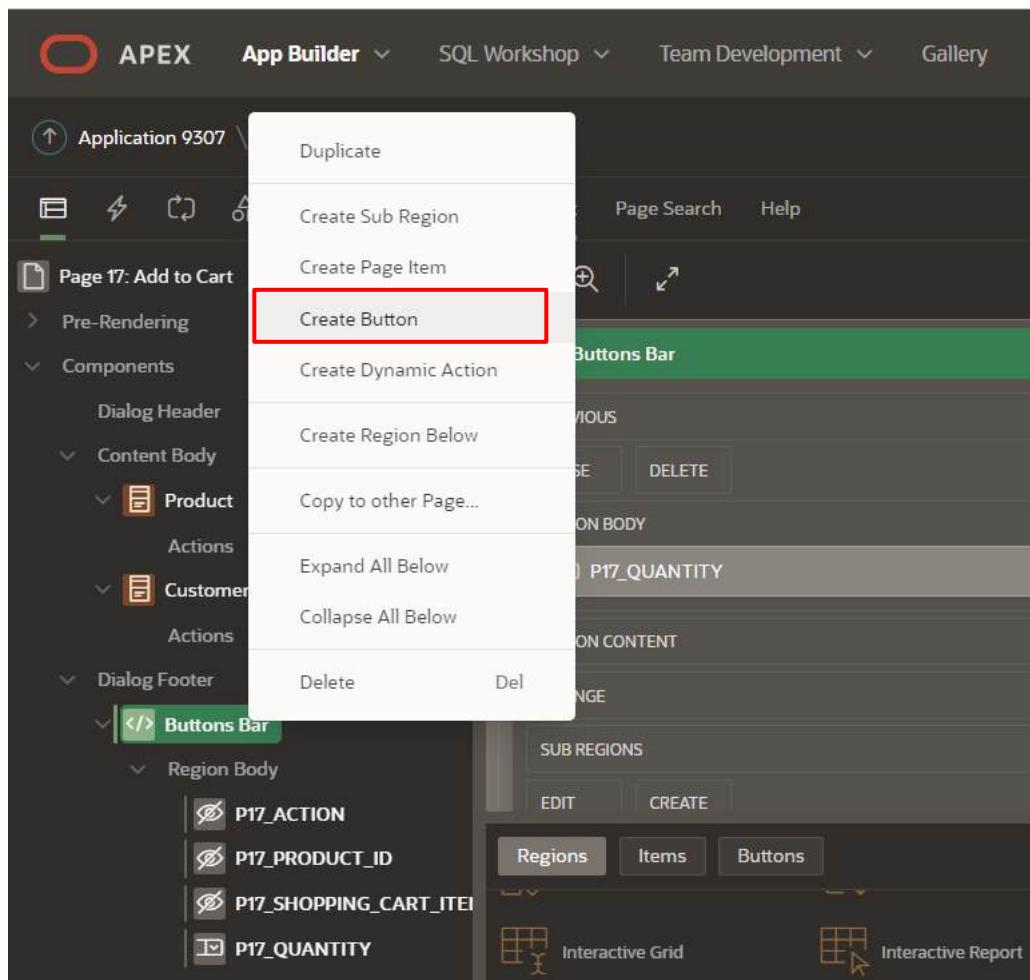
| Display Value | Return Value |
|---------------|--------------|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |

- Click **OK**.
- Set Display Extra Values to **Off**.
- Set Display Null Value to **Off**.



7. Navigate to the **Buttons Bar** region (left side).

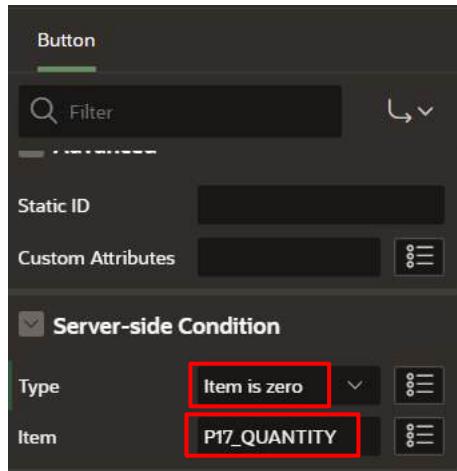
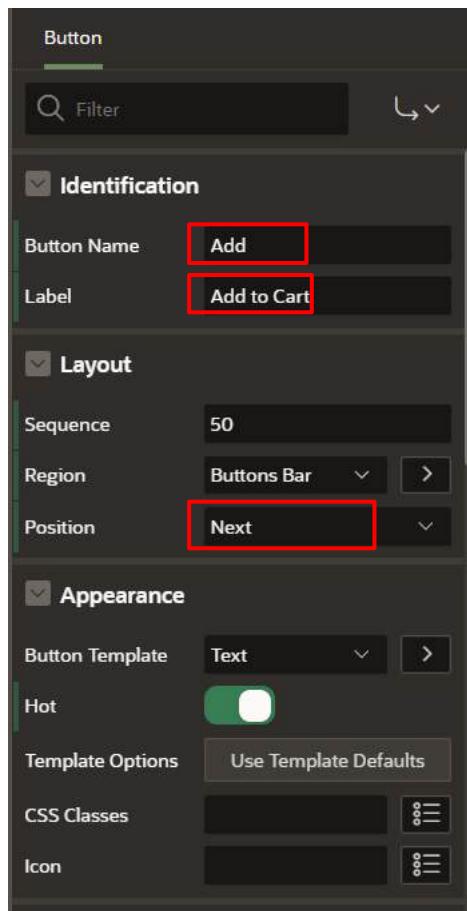
8. Right-click the region and select **Create Button**.



9. Create three buttons as follows:

Table 6: Create Application Page Controls | Practice 1: Add Items and Buttons to a Page

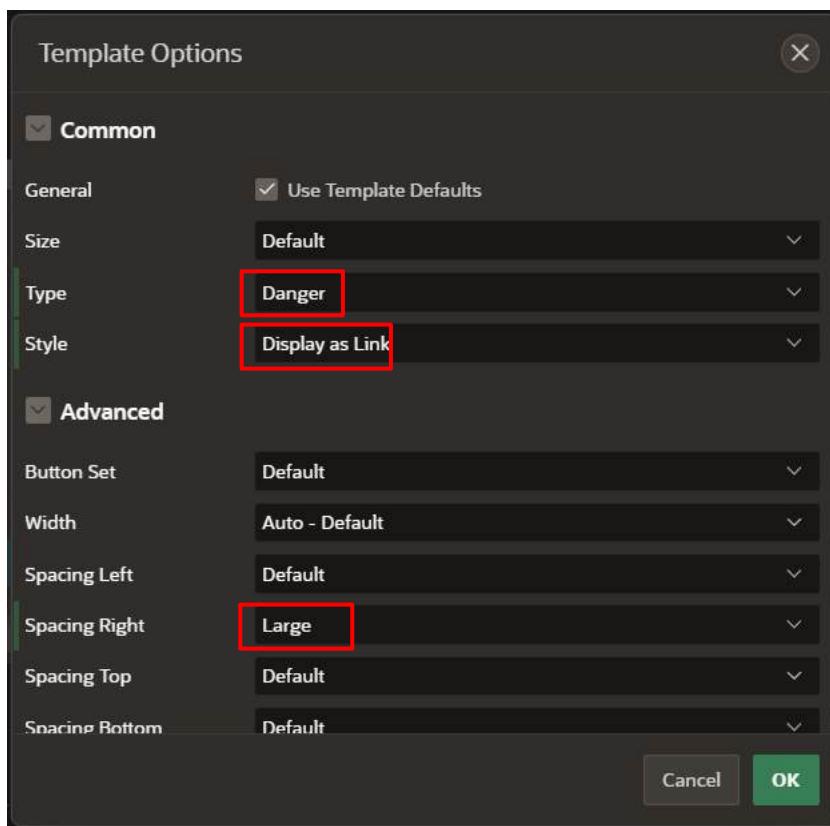
| Name | Label | Button Position | Button Template | Hot |
|--------|------------------|-----------------|-----------------|-----|
| Add | Add to Cart | Next | Text | On |
| Edit | Update Quantity | Create | Text | On |
| Delete | Remove from Cart | Edit | Text | Off |



10. For the **Delete** button, apply the following changes:

- Under Appearance, click Template Options:
 - Type: **Danger**
 - Style: **Display as Link**
 - Spacing Right: **Large**
- Click **OK**.

- Click **Save**.



You now know how to create page items and page buttons. You may now **proceed to the next practice**.



Practice: Enhancing your application using Computations, Processes and Validations

Practice 1: Add Branches, Validations, and Processes to the Shopping Cart Page

Overview

This Hands-on Lab is a collection of six tasks. After completing this lab, your application will enable customers to:

- Review the items in the shopping cart
- Edit the quantity of the items
- Remove an item
- Clear the shopping cart
- Proceed to checkout

In this lab, you will:

- Create Validations, Processes, and Branches to manage the Shopping Cart

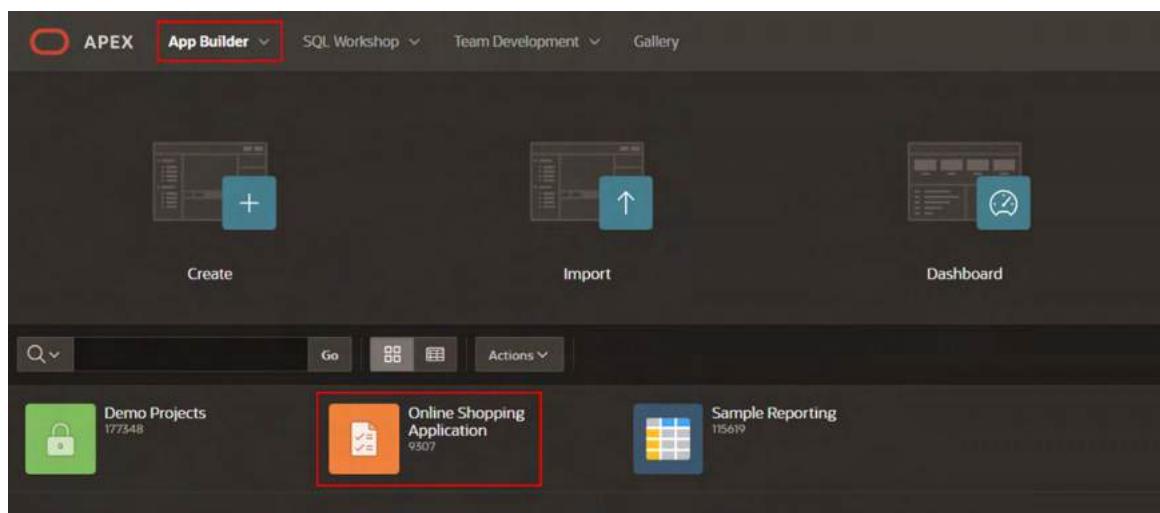
Downloads

- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SOL Workshop](#) workshops.

Tasks

Create Validations on the Page

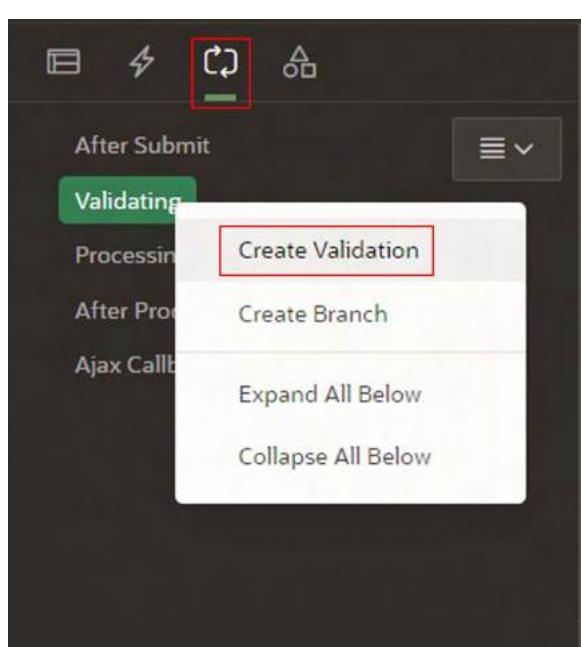
1. Navigate to the **App Builder**. Then, click **Online Shopping Application**.



2. Now, you select **Shopping Cart** under **Page Icons**.



3. In the Rendering tree (left pane), click the **Processing** tab.
4. Over **Validating**, right-click **Create Validation**.



5. Create three validations for the following items: Name, Email, and Store

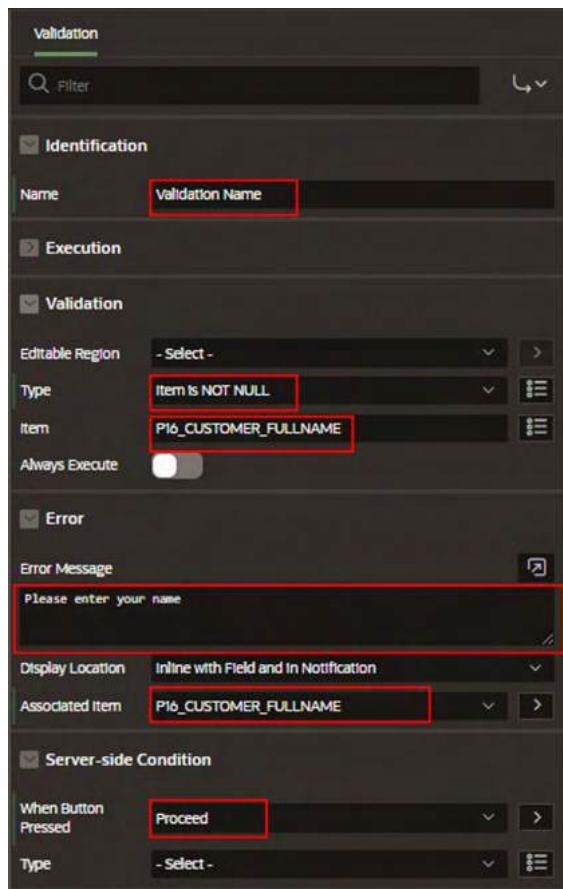


Table 1: Enhance your application using Computations, Processes, and Validations | Lab 1:
Add Branches, Validations, and Processes to the Shopping Cart Page

| Name | Type (under Validation) | Item |
|------------------|-------------------------|-----------------------|
| a.Validate Name | Item is NOT NULL | P16_CUSTOMER_FULLNAME |
| b.Validate Email | Item is NOT NULL | P16_CUSTOMER_EMAIL |
| c.Validate Store | Item is NOT NULL | P16_STORE |

Under Error:

Table 2: Enhance your application using Computations, Processes, and Validations | Lab 1: Add Branches, Validations, and Processes to the Shopping Cart Page

| Error Message | Display Location | Associated Item |
|---------------------------------|---------------------------------------|-----------------------|
| Please enter your name | Inline with Field and in Notification | P16_CUSTOMER_FULLNAME |
| Please enter your email address | Inline with Field and in Notification | P16_CUSTOMER_EMAIL |
| Please select a store | Inline with Field and in Notification | P16_STORE |

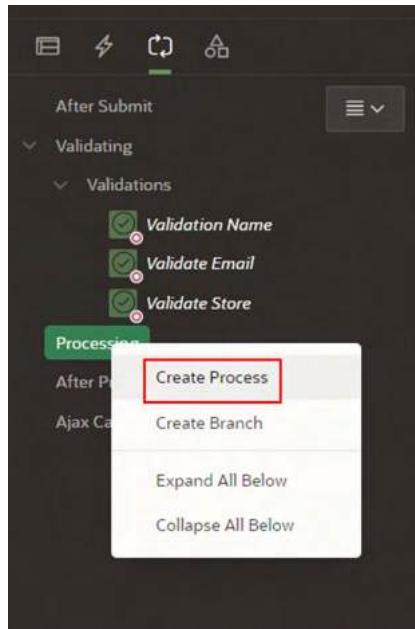
Because these validations only apply when the user proceeds to checkout, let's create that condition. Under Server-side Conditions, set the following:

Table 3: Enhance your application using Computations, Processes, and Validations | Lab 1: Add Branches, Validations, and Processes to the Shopping Cart Page

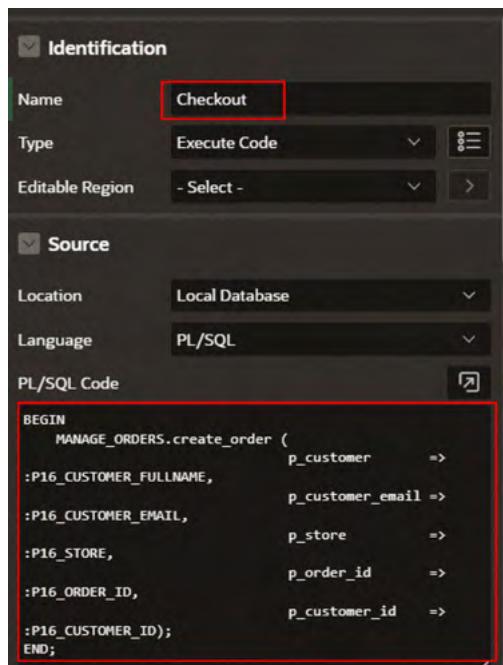
| Name | When Button Pressed |
|----------------|---------------------|
| Validate Name | Proceed |
| Validate Email | Proceed |
| Validate Store | Proceed |

Add a Process to Create the Order

1. On the **Processing** tab (left pane), right-click **Processing** and click **Create Process**.



2. In the Property Editor, enter the following:



- For Name - enter **Checkout**.
- For Type -select **Execute Code**.
- For PL/SQL Code - enter the following PL/SQL code:

```

BEGIN
    MANAGE_ORDERS.create_order (
        :P16_CUSTOMER_FULLNAME,
        :P16_CUSTOMER_EMAIL,
        :P16_ORDER_ID,
        :P16_CUSTOMER_ID);
END;

```

- For Success Message, enter **Order successfully created: &P16_ORDER_ID.**
- Under Server-side Condition, for When Button Pressed, select **Proceed**.



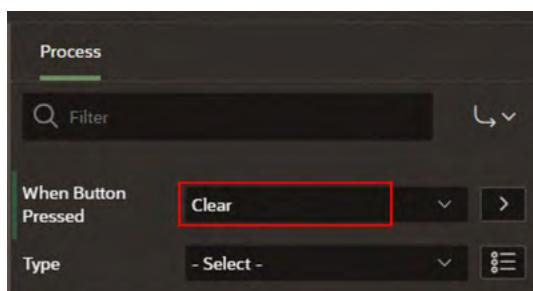
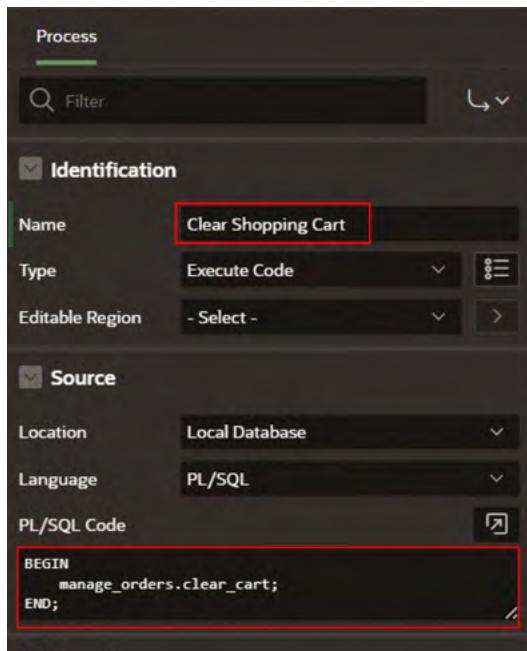
Add Process to Clear the Shopping Cart

1. On the **Processing** tab (left pane), right-click **Processing** and click **Create Process**.
2. Create a second process to clear the shopping cart. In the Property Editor and enter the following:
 - For Name - enter **Clear Shopping Cart**.
 - For Type - select **Execute Code**.

- For PL/SQL Code - enter the following PL/SQL code:

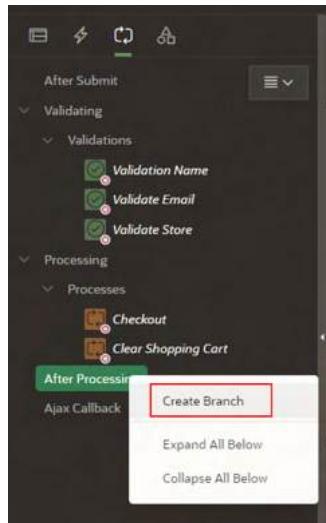
```
BEGIN
    manage_orders.clear_cart;
END;
```

- Under Server-side Condition, for When Button Pressed, select **Clear**.



Add Branches to the Page

1. On the **Processing** tab (left pane), right-click **After Processing** and click **Create Branch**.



2. In the Property Editor, enter the following:

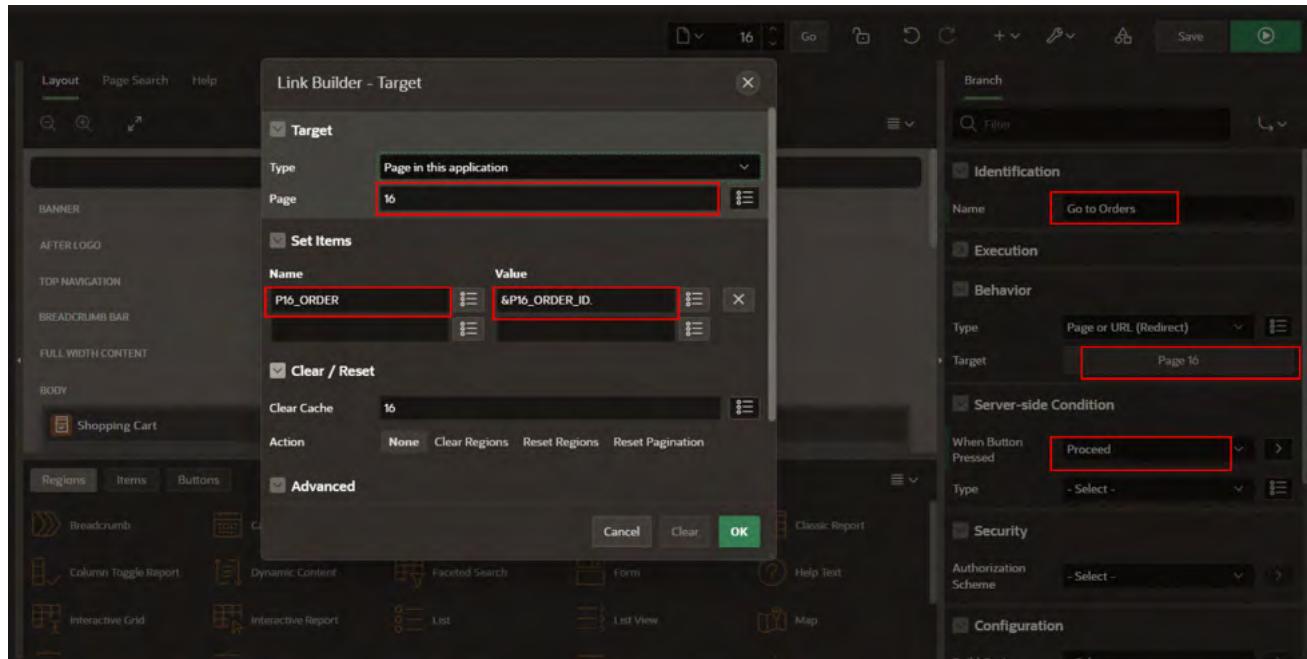
- For Name - enter **Go to Orders**.
- Navigate to Target attribute and click **No Link Defined**.
 - For Type - select Page in this application.
 - For Page - enter **16**.
 - For Set Items - enter:

Table 4: Enhance your application using Computations, Processes, and Validations | Lab 1: Add Branches, Validations, and Processes to the Shopping Cart Page

| Name | Value |
|-----------|----------------|
| P16_ORDER | &P16_ORDER_ID. |

- For Clear Cache - enter **16**.
- Click **OK**.

- Under Server-side Condition, for When Button Pressed, select **Proceed**.



- Create a second branch when the user clears the shopping cart. Right-click **After Processing** and click **Create Branch**.
- In the Property Editor, enter the following:
 - For Name - enter **Go to Products**.
 - Navigate to the Target attribute and click **No Link Defined**.
 - For Type - select Page in this application.
 - For Page - enter **1**.
 - For Clear Cache - enter **1**.
 - Click **OK**.
 - Under Server-side condition, for When Button Pressed, select **Clear**.

You now know how to add validations, processes, and branches to your APEX page. You may now **proceed to the next practice**.

Practice: Creating and using Dynamic Actions

Practice 1: Add Dynamic Actions, Computations, and Processes

Overview

This practice is a collection of nine tasks. After completing this lab, your application will enable customers to:

- Create dynamic actions to manage the Shopping Cart
- Review the product details
- Add, edit, or remove the product from the shopping cart
- Read the customer reviews

In this lab, you will:

- Create dynamic actions for the Shopping Carts Page
- Add Computations and Page Process to the Add to Cart page

Downloads

- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#), [Using SQL Workshop](#), and Enhancing your application using Computations, Processes, and Validations workshops.

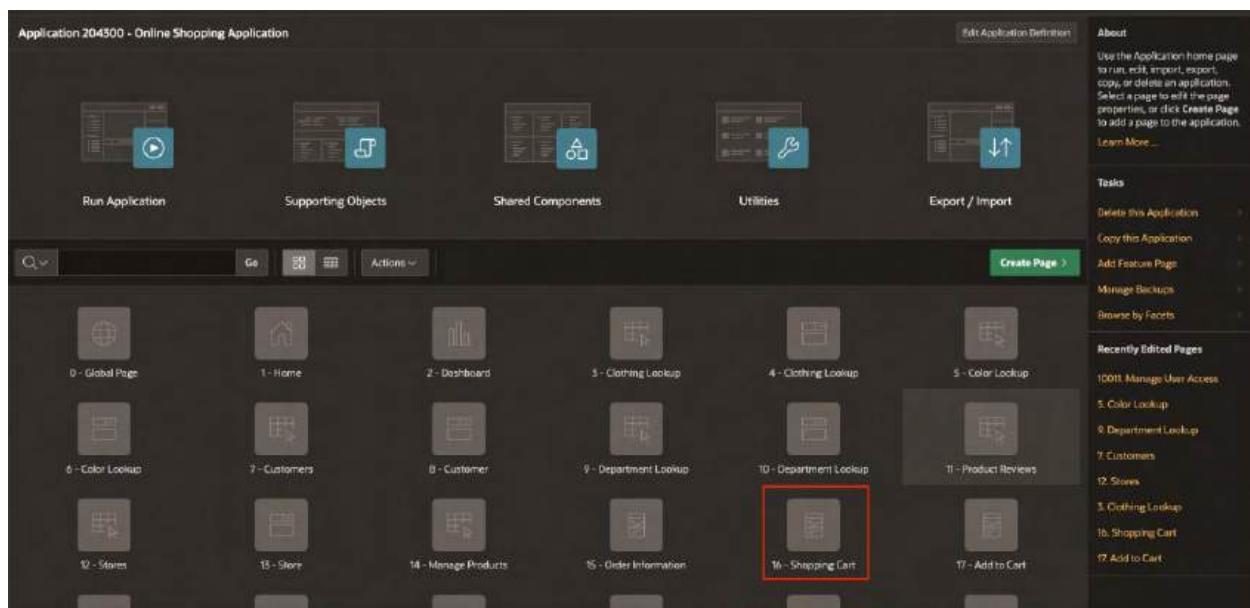
Tasks

Add Dynamic Actions

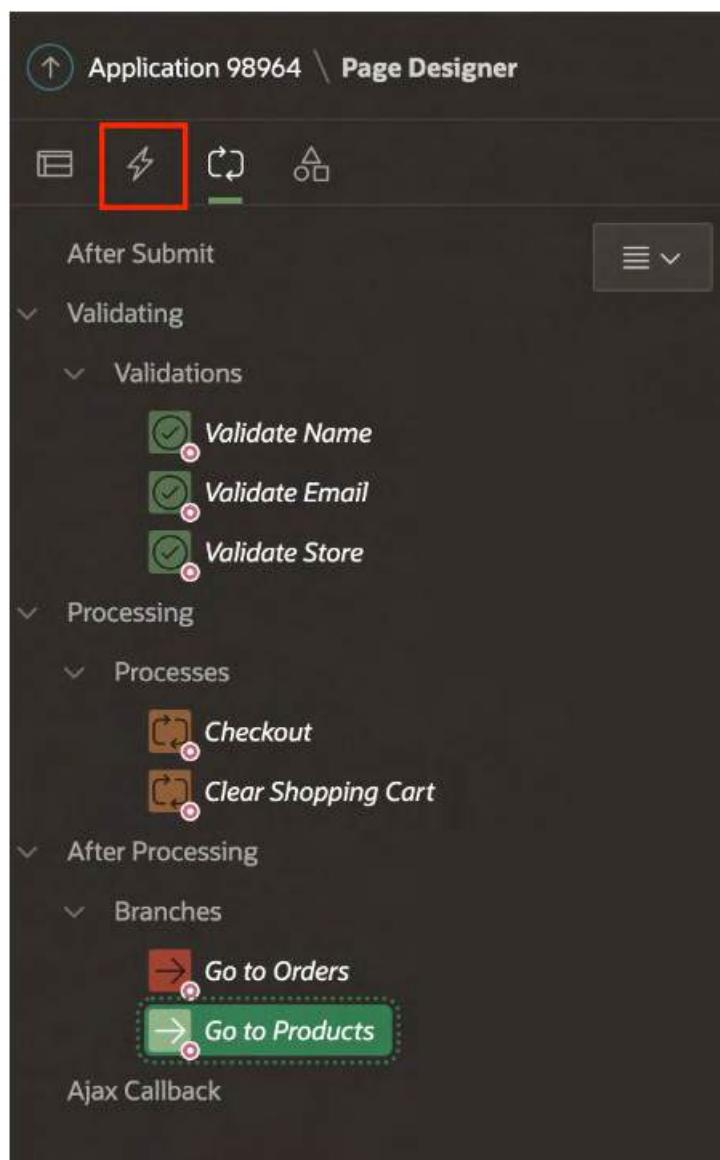
In this task, you will create dynamic actions to:

- Update the badge and icon shown in the navigation bar after the customer has added/edited/removed a product from the shopping cart
- Refresh the shopping cart region

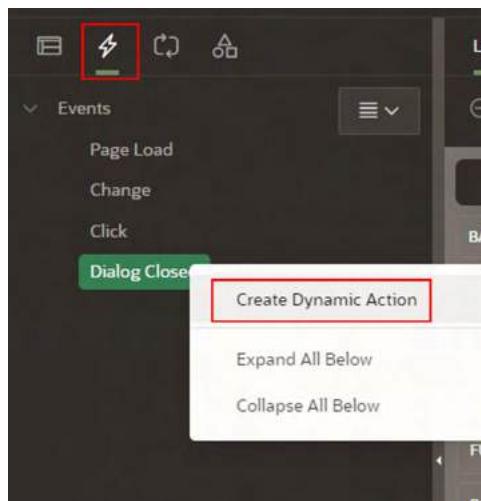
1. Go to **Shopping Cart** page (Page No 16) in the page designer.



2. Navigate to the **Dynamic Actions** tab (left pane).



3. Right-click **Dialog Closed** and select **Create Dynamic Action**.



4. In the Property Editor, enter the following:

- Under Identification section:
 - For Name - enter **Update Shopping Cart Header**.
- Under When section:
 - For Event - select **Dialog Closed**.
 - For Selection Type - select **Region**.
 - For Region - select **Shopping Cart**.
- Under Client-side Condition:
 - For Type - select **JavaScript expression**.
 - For JavaScript Expression, enter the following:

```
parseInt(this.data.P17_SHOPPING_CART_ITEMS) > 0
```

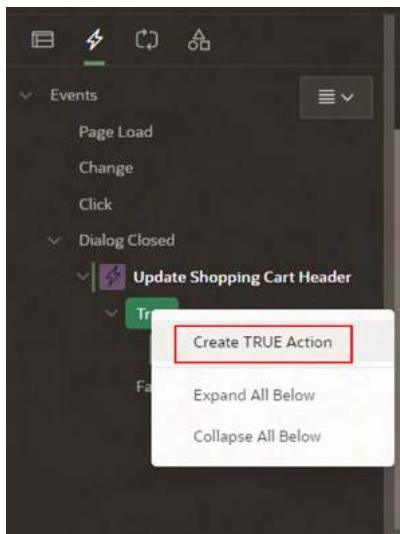
5. Navigate to **Refresh** Action.

- Under Identification section:
 - For Name - enter **Update Badge & Icon**
 - For Action - select **Execute JavaScript Code**.
- Under Settings section:
 - For Code - enter the following JavaScript Code:

```
// Update Badge Text  
apex.jQuery(".js-shopping-cart-item .t-Button-badge").text(this.data.P17_SHOPPING_CART_ITEMS);  
  
// Update Icon  
apex.jQuery(".js-shopping-cart-item .t-Icon").removeClass('fa-cart-empty').addClass('fa-cart-full');
```

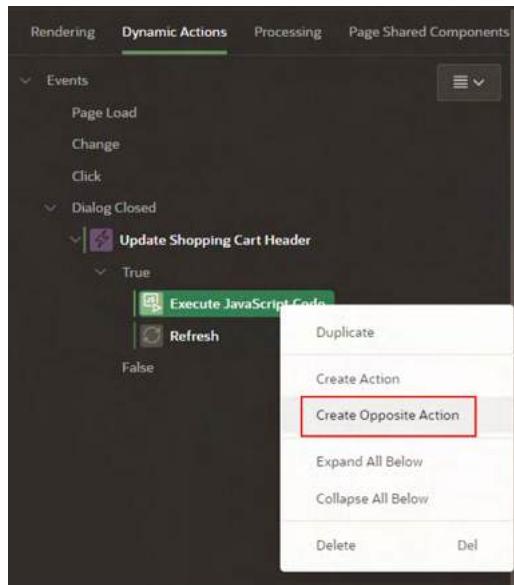


6. Create a second action. In the Dynamic Actions tab (left pane), navigate to **True** under the **Update Shopping Cart Header Dynamic Action**, right-click True, and select **Create TRUE Action**



7. In the Property Editor, enter the following:
 - Under Identification section:
 - For Name - enter Refresh Shopping Cart region
 - For Action - select **Refresh**.
 - Under Affected Elements section:
 - For Selection Type - select **Region**.
 - For Region - select **Shopping Cart**.
8. Create an opposite action. In the Dynamic Actions tab (left pane), navigate to the **Execute JavaScript Code** action.

9. Right-click **Execute JavaScript Code** and select **Create Opposite Action**.

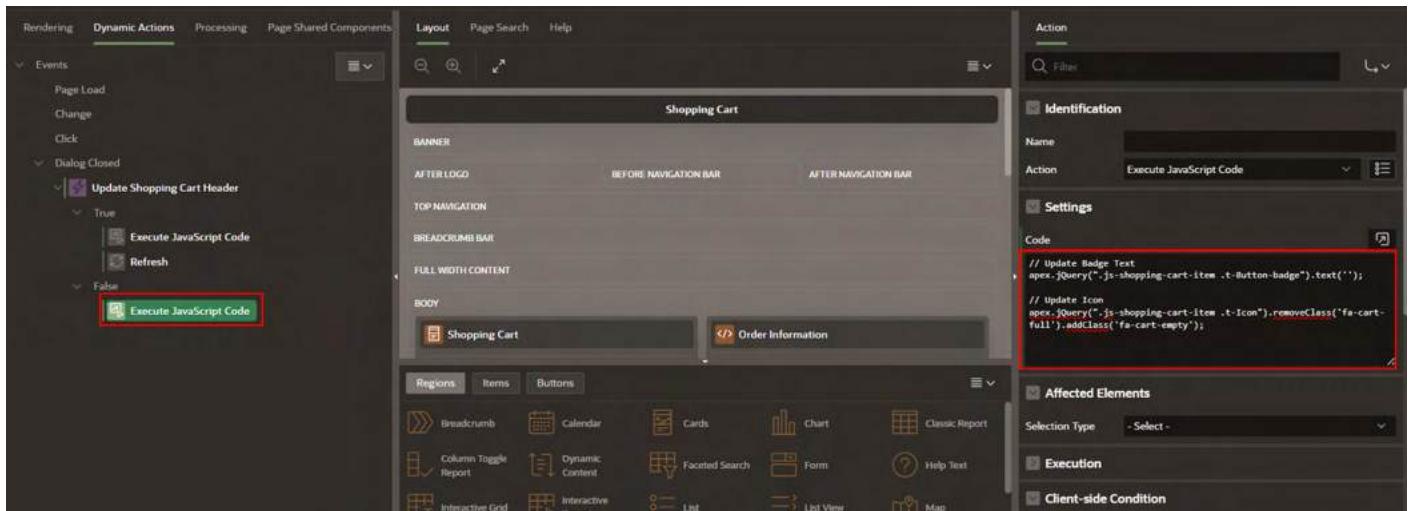


10. Navigate to the **Execute JavaScript Code** action under the False heading.

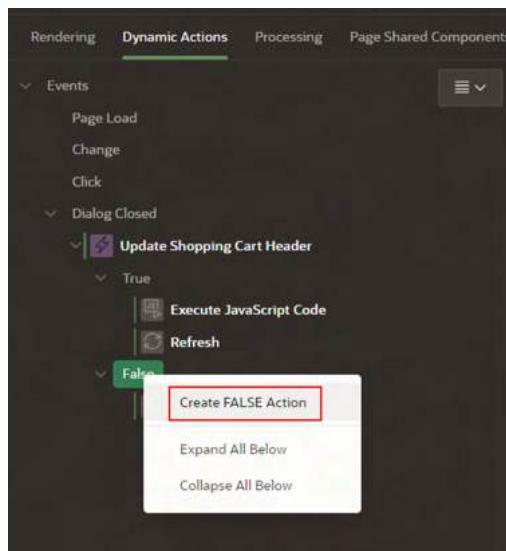
- Under Identification section:
 - For Action - select Execute JavaScript Code.
- Under Settings section:
 - For Code - enter the following JavaScript Code:

```
// Update Badge Text
apex.jQuery(".js-shopping-cart-item .t-Button-badge").text(' ');

// Update Icon
apex.jQuery(".js-shopping-cart-item .t-Icon").removeClass('fa-cart-full').addClass('fa-cart-empty');
```



11. Create a second action. In the Dynamic Actions tab (left pane), navigate to **False** under the **Update Shopping Cart Header** dynamic action.



12. In the Property Editor, enter the following:

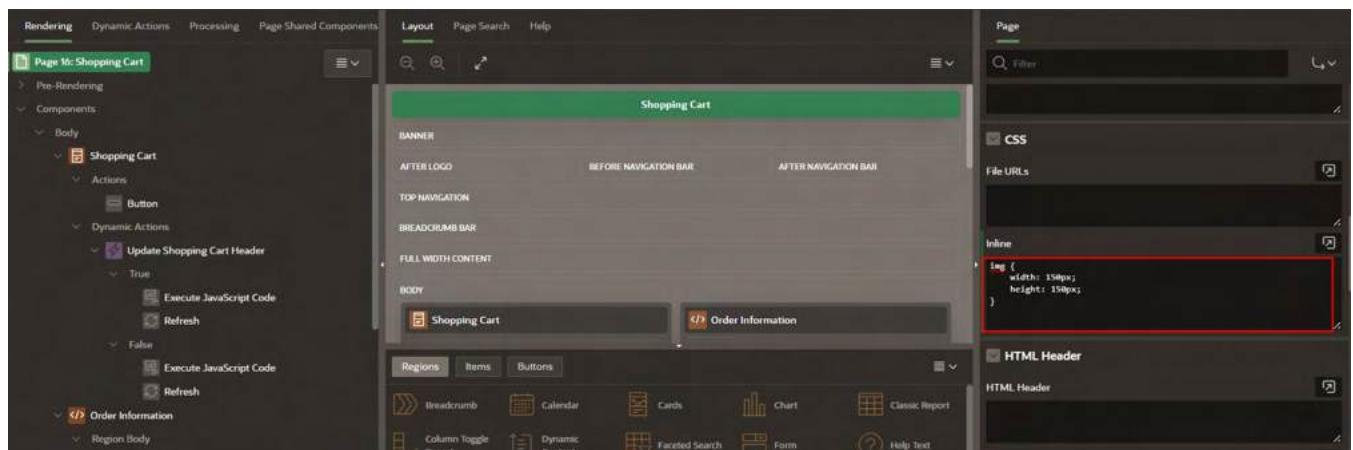
- Under Identification section:
 - For Name - enter Refresh Shopping Cart region
 - For Action - select **Refresh**.
- Under Affected Elements section:
 - For Selection Type - select **Region**.
 - For Region - select **Shopping Cart**.

13. Click **Save**.

Format Products Image Size

1. In the Rendering tree (left pane), navigate to **Page 16: Shopping Cart**.
2. In the Property Editor (right pane), do the following:
 - Under CSS section:
 - For Inline - enter the following:

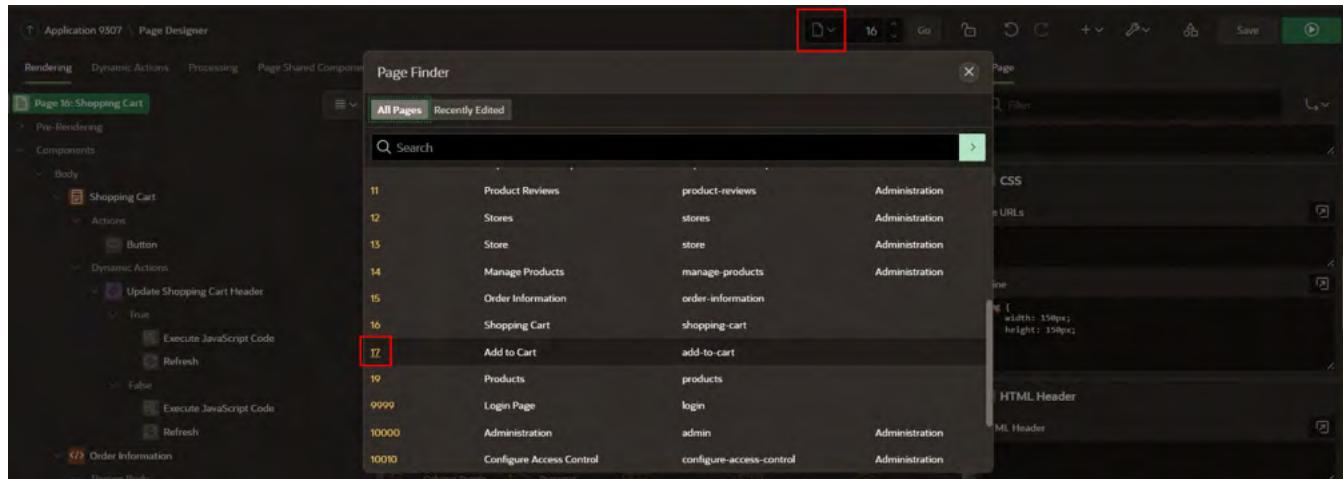
```
img {  
    width: 150px;  
    height: 150px;  
}
```



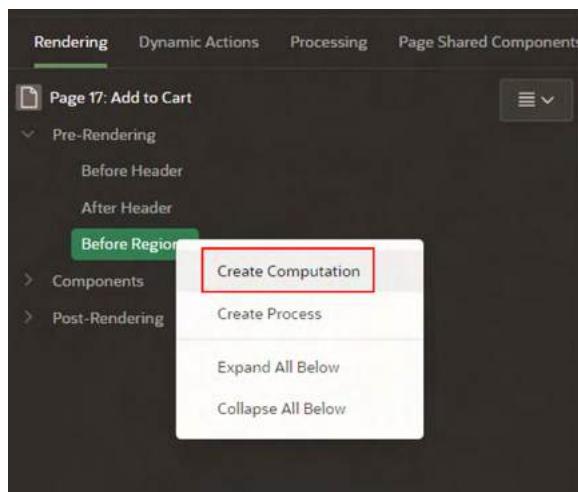
3. Click **Save**.

Add Computation to Calculate the Number of Items for a Product

1. Navigate to **Page Finder** and click **File symbol**. Then in the popup **Page Finder**, select page **17**.



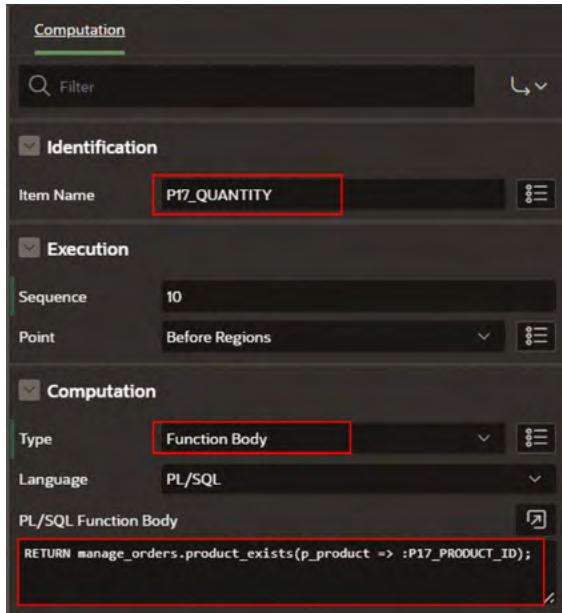
2. In the Rendering tree (left pane), expand the **Pre-Rendering**.
3. Right-click **Before Regions** and select **Create Computation**.



4. In the Property Editor, enter the following:
 - Under Identification section:
 - For Item Name - select **P17_QUANTITY**.
 - Under Computation:
 - For Type - select **Function Body**.

- For PL/SQL Function Body - enter the following PL/SQL Code:

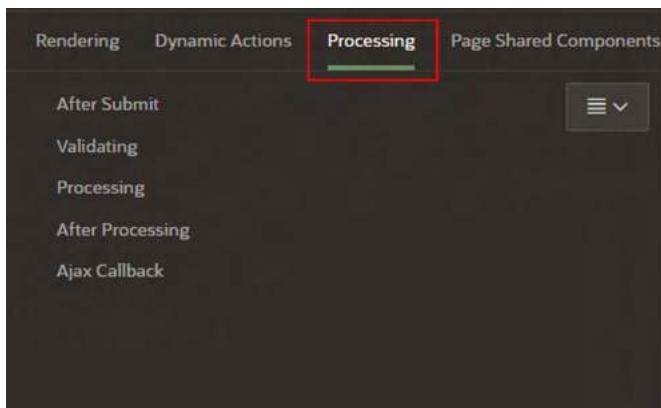
```
RETURN manage_orders.product_exists(p_product =>
:P17_PRODUCT_ID);
```



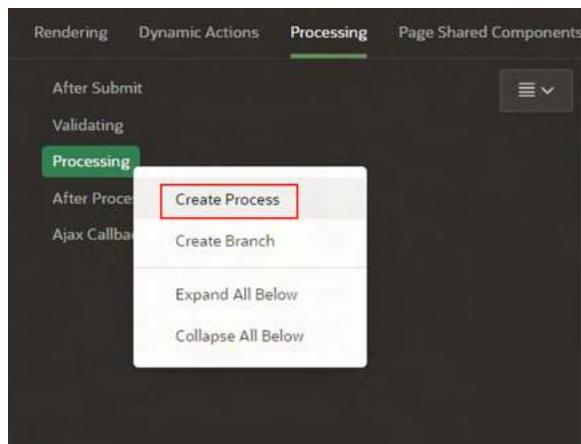
Create a Process to Add Products to the Shopping Cart

In this Task, you call the *manage_orders.add_product* procedure that will add a product temporarily to the APEX collection.

1. In the Rendering tree (left pane), navigate to the **Processing** tab.



2. Right-click **Processing** and select **Create Process**.



3. In the Property Editor, enter the following:

- For Name - enter **Add product**.
- For Type - select **Execute Code**.
- For PL/SQL Code - enter the following code:

```
BEGIN
    IF manage_orders.product_exists(p_product =>
:PI7_PRODUCT_ID) = 0 THEN
        manage_orders.add_product (p_product =>
:PI7_PRODUCT_ID,
                                p_quantity => :PI7_QUANTITY);
    END IF;
    :PI7_ACTION := 'ADD';
END;
```

The screenshot shows the Oracle ADF Faces interface with the 'Processing' tab selected. In the center, there's a dialog box titled 'Add to Cart' with a 'Product' region. On the right, the 'Process' editor is open for the 'Add product' process. The 'Identification' section has 'Name' set to 'Add product' and 'Type' set to 'Execute Code'. The 'Source' section shows the code is located in 'Local Database' using 'PL/SQL'. The 'PL/SQL Code' pane contains the provided PL/SQL block.

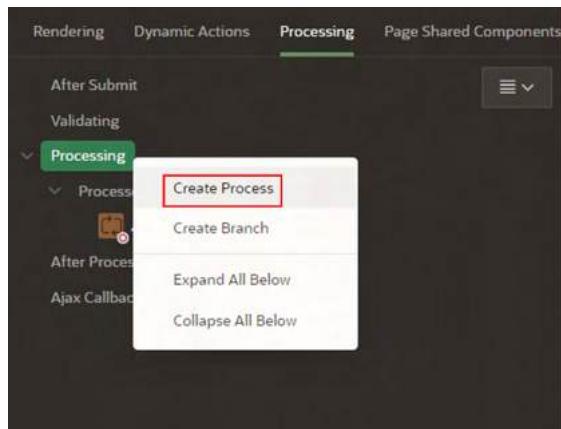
- Under the Server-side Condition section:
 - For When Button Pressed - select **Add**.

4. Click **Save.**

Create a Process to Edit Products in the Shopping Cart

In this Task, you call the *manage_orders.remove_product* and *manage_orders.add_product* procedures to remove a product from the shopping cart and add it again with the updated quantity.

1. In the **Processing tab, right-click **Processing** and select **Create Process**.**



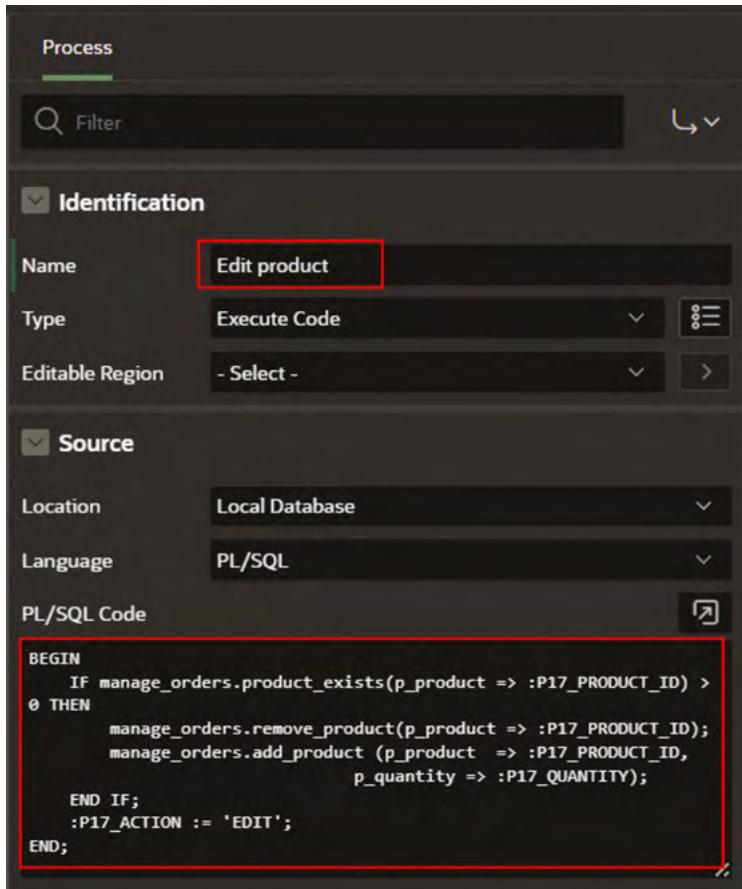
2. In the Property Editor, enter the following:

- For Name - enter **Edit product**.
- For Type - select **Execute Code**.
- For PL/SQL Code - enter the following PL/SQL code:

```
BEGIN
    IF manage_orders.product_exists(p_product =>
:P17_PRODUCT_ID) > 0 THEN
        manage_orders.remove_product(p_product =>
:P17_PRODUCT_ID);
        manage_orders.add_product (p_product =>
:P17_PRODUCT_ID,
                                p_quantity => :P17_QUANTITY);
    END IF;
    :P17_ACTION := 'EDIT';
END;
```

- Under Server-side Condition section:
 - For When Button Pressed, select **Edit**.

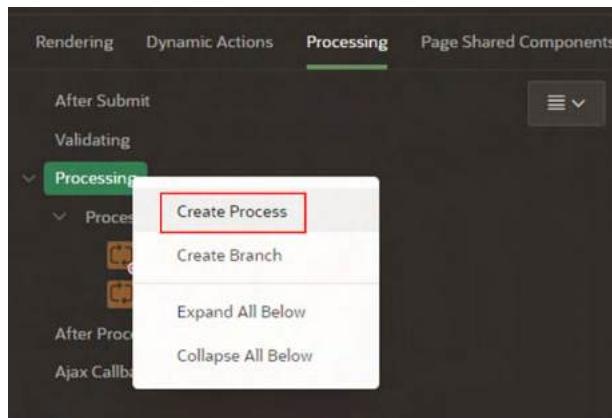
Click **Save**.



Create a Process to Delete Products from the Shopping Cart

In this Task, you call the `manage_orders.remove_product` to remove a product from the shopping cart.

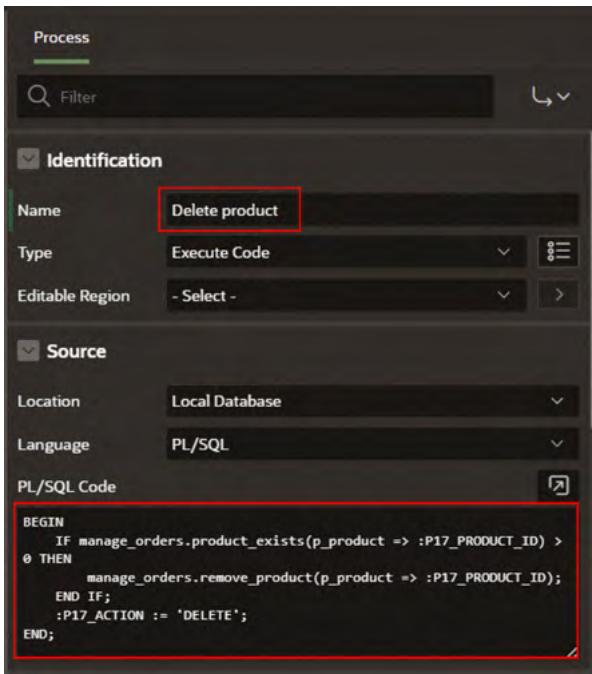
1. In the **Processing** tab, right click **Processing** and select **Create Process**.



In the Property Editor, enter the following:

- For Name - enter `Delete_product`.
- For Type - select **Execute Code**.
- For PL/SQL Code - enter the following PL/SQL code:

```
BEGIN
    IF manage_orders.product_exists(p_product =>
:P17_PRODUCT_ID) > 0 THEN
        manage_orders.remove_product(p_product =>
:P17_PRODUCT_ID);
    END IF;
    :P17_ACTION := 'DELETE';
END;
```



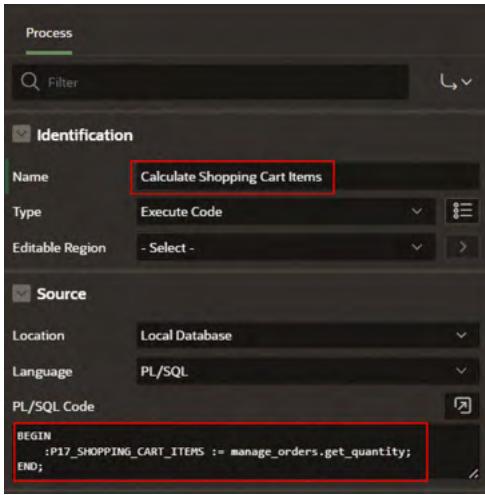
- Under the Server-side Condition section:
 - For **When Button Pressed**, select **Delete**.
- Click **Save**.

Create a Process to Calculate the Shopping Cart Items

In this task, you call the `manage_orders.get_quantity` to get the total number of products in the shopping cart.

1. In the **Processing** tab, right-click **Processing** and select **Create Process**.
2. In the Property Editor, enter the following:
 - For Name - enter **Calculate Shopping Cart Items**.
 - For Type - select **Execute Code**.
 - For PL/SQL Code - enter the following PL/SQL code:

```
BEGIN
    :P17_SHOPPING_CART_ITEMS := manage_orders.get_quantity;
END;
```



Click **Save**.

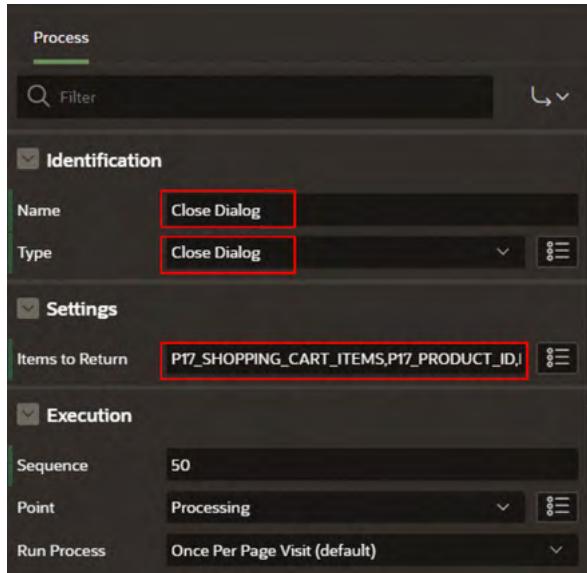
Create a Process to Close the Modal Page

After the customer has taken an action (add/edit/delete) regarding the product, the modal page will close and the shopping process continues.

1. In the **Processing** tab, right-click **Processing** and click **Create Process**.
2. In the Property Editor, enter the following:
 - Under the Identification section:
 - For Name - enter **Close Dialog**.
 - For Type - select **Close Dialog**.

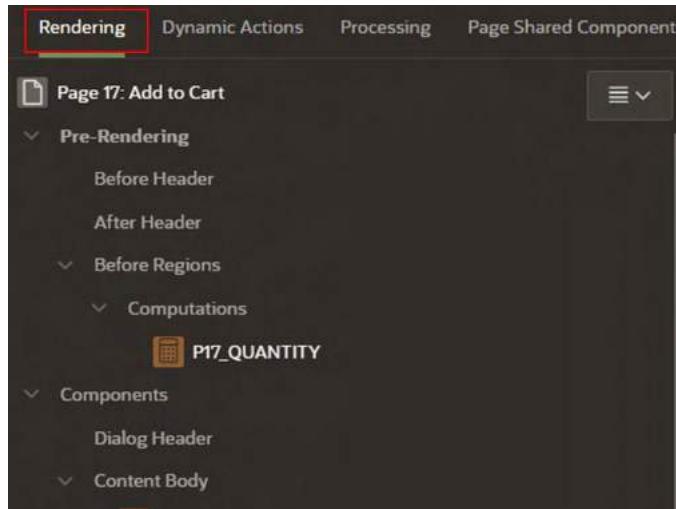
- Under Settings section:
 - For Items to Return - enter
P17_SHOPPING_CART_ITEMS,P17_PRODUCT_ID,P17_ACTION,P17_QUANTITY.

Click **Save**.



Enhance the Modal Page

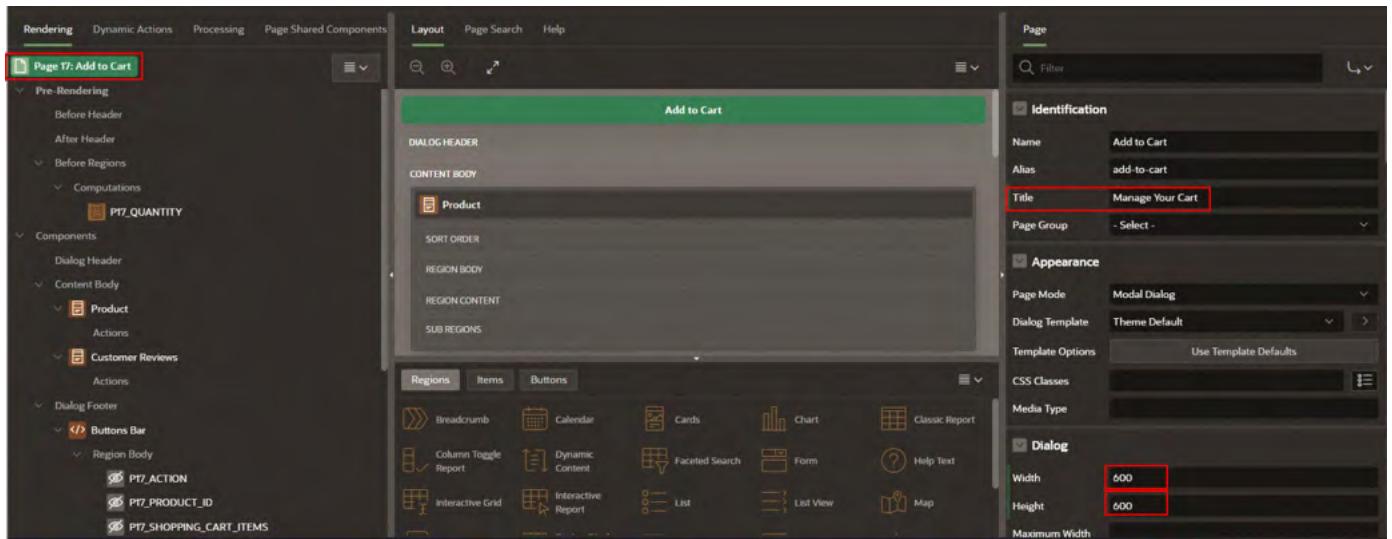
1. Navigate to the **Rendering** tab (left pane).



2. In the rendering tree, select the root node in the component tree **Page 17: Add to Cart**.

3. In the Property Editor, make the following changes:

- Under the Identification section: For Title, enter **Manage Your Cart**.
- Under the Dialog section:
 - For Width, enter **600**.
 - For Height, enter **600**.



4. Click **Save**.

You now know how to customize and enhance the APEX page. You may now **proceed to the next practice**.

Practice: Managing Cards, Faceted Search, and Smart Filters

Practice 1: Improve Smart Filters Page

Overview

In this lab, you will gain an insight into the abilities of Smart Filters, and learn how to add **New Filters** and Edit existing Filters

Downloads

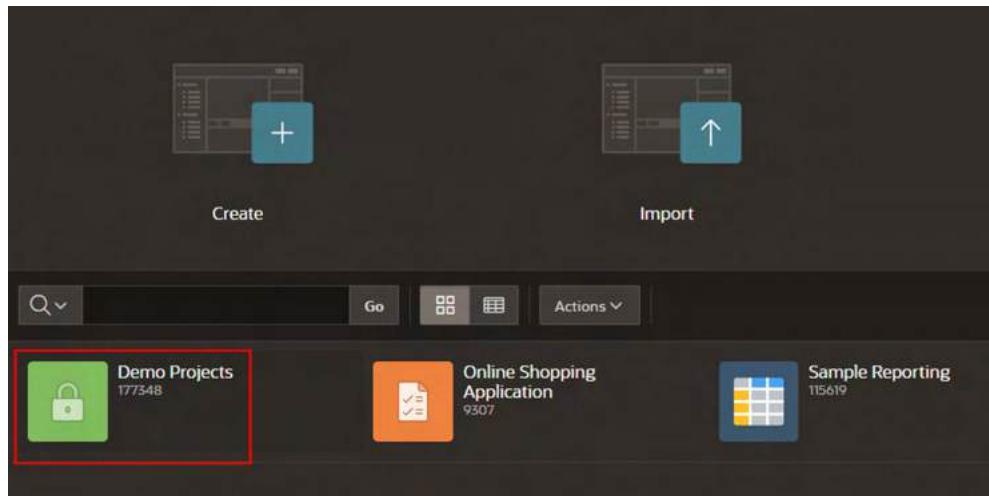
- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SOL Workshop](#) workshops.

Task

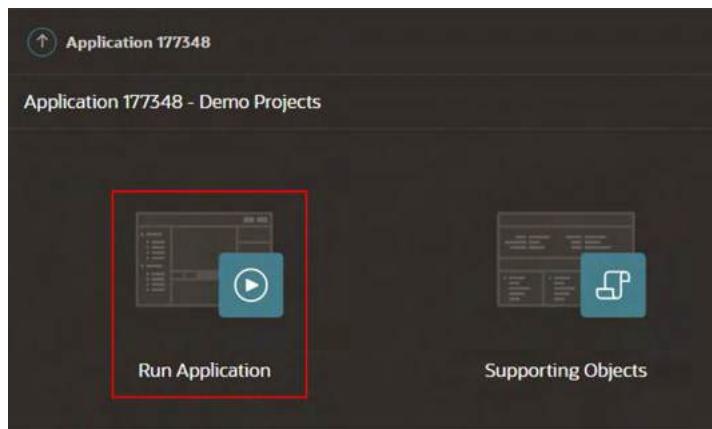
Enhance Smart Filters

In this Lab, you are going to customize the **Smart Filters** page in the **Demo Projects** application. The **Project Tasks Search** page utilizes Smart Filters to display the records.

1. Navigate to the **App Builder**. Then, click the **Demo Projects** application.



- Now, you run the application using the **Run Application** icon.



- Navigate to the **Project Tasks Search** page and click **Page 8** in the **Developer toolbar**. Note that your page number might be different.

The screenshot shows the "Project Tasks Search" page. The left sidebar has navigation links: Home, Dashboard, Project Tasks, and Project Tasks Search (which is selected). The main area displays a table of project tasks with columns: Project, Task Name, Start Date, End Date, Status, Assigned To, Cost, and Budget. A search bar at the top allows filtering by project name, assignee, budget, and status. The developer toolbar at the bottom includes icons for Home, App:177348, Page 8 (which is highlighted with a red box), Session, Debug, Quick Edit, Customize, and Context. The table data is as follows:

| Project | Task Name | Start Date | End Date | Status | Assigned To | Cost | Budget |
|--------------------------|--|------------|------------|---------|---------------|-------|--------|
| ACME Web Configuration | Identify server requirements | 11/19/2021 | 12/6/2021 | Closed | John Watson | 500 | 300 |
| Maintain Support Systems | HR software upgrades | 12/4/2021 | 1/1/2022 | On-Hold | Pam King | 8,000 | 9,000 |
| Maintain Support Systems | Apply Billing System updates | 1/7/2022 | 1/23/2022 | On-Hold | Russ Sanders | 9,500 | 7,000 |
| ACME Web Configuration | Determine Web listener configuration(s) | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| ACME Web Configuration | Specify security authentication scheme(s) | 1/2/2022 | 1/4/2022 | Closed | John Watson | 200 | 300 |
| ACME Web Configuration | Select servers for Development, Test, Production | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| Email Integration | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3,000 | 1,500 |
| ACME Web Configuration | Configure Workspace provisioning | 12/20/2021 | 1/4/2022 | Closed | John Watson | 200 | 100 |
| ACME Web Configuration | Create pilot workspace | 1/15/2022 | 2/8/2022 | Closed | John Watson | 100 | 100 |
| ACME Web Configuration | Run installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| Bug Tracker | Implement bug tracking software | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| Bug Tracker | Review automated testing tools | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff | 2,750 | 1,500 |
| Train Developers | Publish development standards | 12/3/2021 | 12/13/2021 | On-Hold | John Watson | 1,000 | 2,000 |
| Train Developers | Publish links to self-study courses | 12/28/2021 | 1/3/2022 | Closed | John Watson | 100 | 100 |

- Now, Update **P8_PROJECT**, Change type to **Radio Group** and enable the **Client-Side Filtering** attribute.

In the Rendering tab, select the P8_PROJECT filter (if not already selected). In the Property Editor, under the **Identification** change type to **Radio Group**, then find **List Entries** and enable **Client-Side Filtering**. Enabling this option renders a search field above the list entries.

- Click **Save and Run Page** in the upper-right corner. If prompted, enter your workspace username and password and click **Sign In**. The revised page appears.

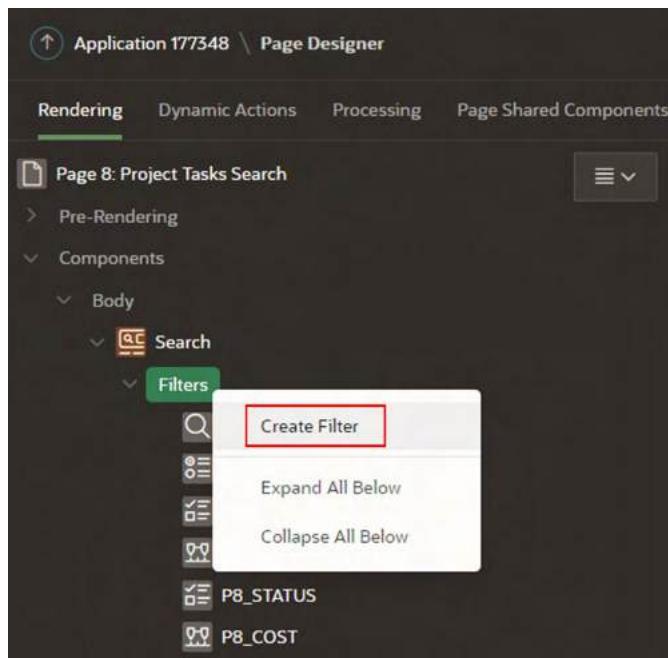
| | Name | Start Date | End Date | Status | Assigned To | Cost | Budget |
|------------------------|-------------------------------------|------------|------------|---------|---------------|-------|--------|
| ACME Project | Configure server requirements | 11/19/2021 | 12/6/2021 | Closed | John Watson | 500 | 300 |
| ACME Project | Install software upgrades | 12/4/2021 | 1/1/2022 | On-Hold | Pam King | 8,000 | 9,000 |
| ACME Project | Update Billing System | 1/7/2022 | 1/29/2022 | On-Hold | Russ Sanders | 9,500 | 7,000 |
| ACME Project | Configure Email Integration | 12/16/2021 | 12/17/2021 | Closed | James Cassidy | 100 | 100 |
| ACME Project | Test Employee Satisfaction Survey | 1/2/2022 | 1/4/2022 | Closed | John Watson | 200 | 300 |
| ACME Project | Deploy Load Packaged Applications | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| ACME Project | Maintain Support Systems | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3,000 | 1,500 |
| ACME Project | Migrate Desktop Application | 12/20/2021 | 1/4/2022 | Closed | John Watson | 200 | 100 |
| ACME Project | Migrate from Legacy Server | 1/15/2022 | 2/8/2022 | Closed | John Watson | 100 | 100 |
| ACME Web Configuration | Run installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| Bug Tracker | Implement bug tracking software | 1/5/2022 | 1/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| Bug Tracker | Review automated testing tools | 11/25/2021 | 11/27/2021 | On-Hold | Myra Sutcliff | 2,750 | 1,500 |
| Train Developers | Publish development standards | 12/3/2021 | 12/13/2021 | On-Hold | John Watson | 1,000 | 2,000 |
| Train Developers | Provide links to self-study courses | 12/28/2021 | 1/3/2022 | Closed | John Watson | 100 | 100 |
| Train Developers | Facilitate workspace provisioning | 1/15/2022 | 1/22/2022 | Closed | James Cassidy | 500 | 700 |

Add New Smart Filter

Add new filters in Page Designer by right-clicking filter and selecting Create Filter from the context menu.

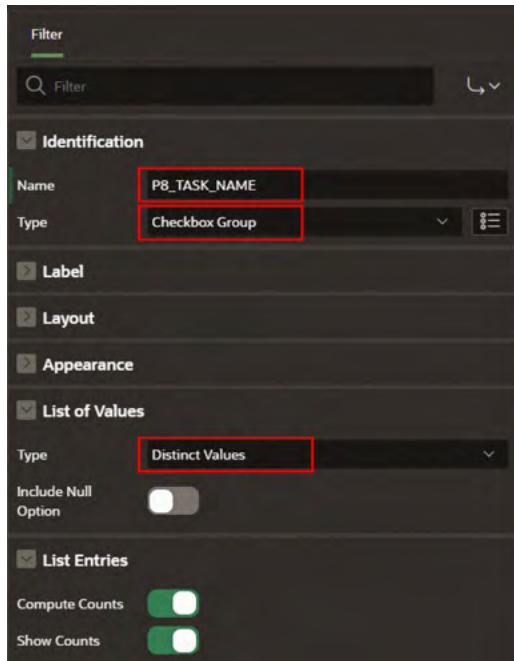
Filters map to specific database column. When creating a new filter, you can create the following types: Checkbox Group, Input Field, Radio Group, and Range.

1. View the smart filters page in Page Designer. In the Rendering tab, right-click Filters and select **Create Filter**.



2. Select the new filter and edit the following filter attributes in the Property Editor:
 - Under Identification:
 - For Name, enter **P8_TASK_NAME**.
 - For Type, select **Checkbox Group**.

- Under **List of Values**:
 - For Type, select **Distinct Values**.



3. Click **Save and Run Page** in the upper-right corner. If prompted, enter your workspace username and password and click Sign In. The revised page appears.

Practice 2: Improve and Customize the Faceted Search and Cards Regions

Overview

In this lab, you will learn how to improve the Products page by adding new facets and customizing the cards.

Please note that customer can quickly identify the products that already have been added to the shopping cart.

In this practice, you will:

- Improve both Faceted Search and Cards region
- Add Dynamic Actions to the page

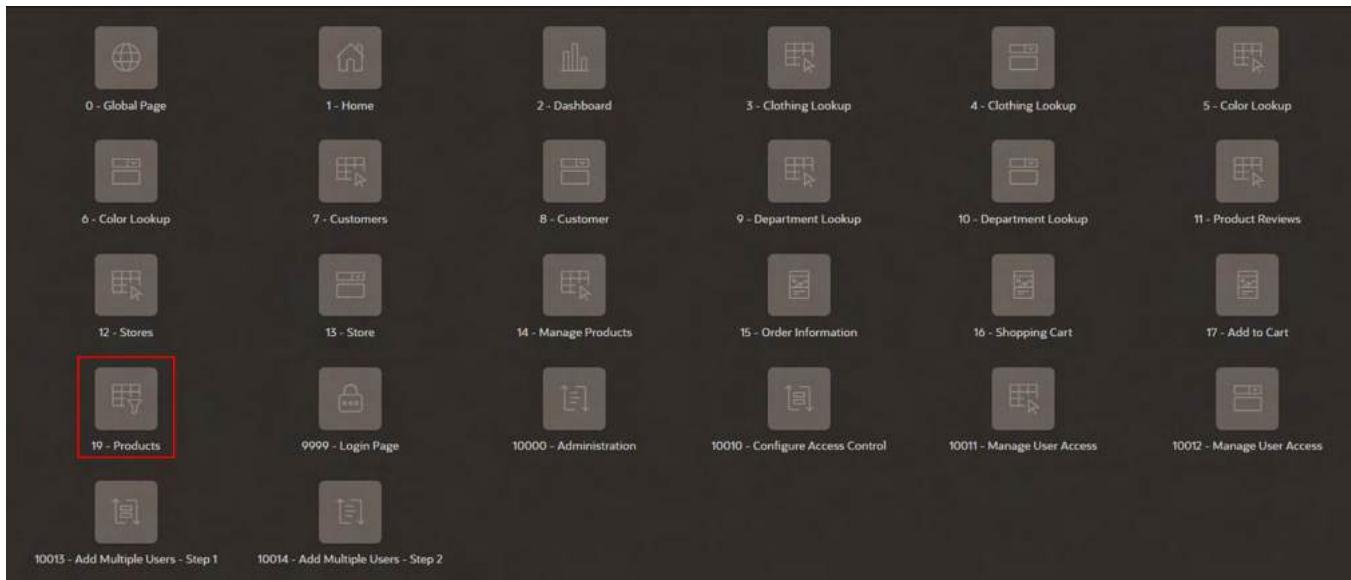
Downloads

- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Task: Navigate to Products Page.

1. From the runtime application, navigate to the **Products** page in **Page Designer**. Given that you run this app from the APEX App Builder, you will find the Developer Toolbar at the bottom of the screen. *{Note: End users who log directly into the app will not see this toolbar.}*
In the Developer Toolbar click **Page 19**.

Alternatively, you can also navigate back to the APEX App Builder tab in your browser manually by selecting the appropriate browser tab or window.
Once in the App Builder, click **19 - Products**.

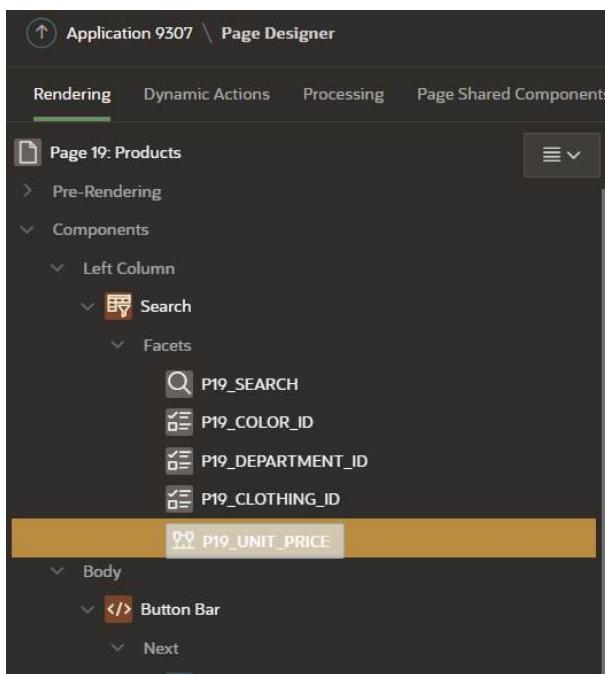


You should now be in Page Designer with **Page 19: Products** loaded.

Reorder Facets

Unit price is not a common search criteria, so you want to put this facet at the bottom.

1. In the Rendering tree (left pane), under Search, within Facets, click and hold **P19_UNIT_PRICE** and drag it down until it is under **P19_CLOTHING_ID**, then release the mouse cursor.



Enhance the Faceted Search

1. In the Rendering tree (left pane), navigate to **Search**.
2. In the Property Editor (right pane), click **Attributes** and do the following:
 - For Total Row Count Label - enter **Total Products**.
 - For Show Charts - select **No**.

The screenshot shows the Oracle APEX Page Designer interface. The left pane displays the rendering tree for 'Page 19: Products', with the 'Search' component under 'Left Column' highlighted. The right pane shows the property editor with the 'Attributes' tab selected. Key settings in the 'Attributes' tab include:

- Total Row Count Label:** Set to **Total Products**.
- Show Charts:** Set to **No**.

Enhance the Cards Region

1. In the Rendering tree (left pane), navigate to **Search Results** and in the Property Editor (right pane), do the following:
 - For SQL Query - enter the following SQL code:

```
SELECT "PRODUCT_ID",
       "PRODUCT_NAME",
       "UNIT_PRICE",
       "PRODUCT_DETAILS",
       "PRODUCT_IMAGE",
       "IMAGE_MIME_TYPE",
       "IMAGE_FILENAME",
       "IMAGE_CHARSET",
       "IMAGE_LAST_UPDATED",
       "COLOR_ID",
       (
           SELECT 11."COLOR"
           FROM   "COLOR_LOOKUP" 11
           WHERE  11."COLOR_ID" = m."COLOR_ID") "COLOR_ID_L$1",
"DEPARTMENT_ID",
(
    SELECT 12."DEPARTMENT"
    FROM   "DEPARTMENT_LOOKUP" 12
    WHERE  12."DEPARTMENT_ID" = m."DEPARTMENT_ID")
"DEPARTMENT_ID_L$2",
"CLOTHING_ID",
(
    SELECT 13."CLOTHING"
    FROM   "CLOTHING_LOOKUP" 13
    WHERE  13."CLOTHING_ID" = m."CLOTHING_ID")
"CLOTHING_ID_L$3",
b.brand
FROM   "PRODUCTS" m,
       json_table (m.product_details, '$' columns ( brand
varchar2(4000) path '$.brand') ) b
```

The screenshot shows the Oracle APEX Page Designer interface. On the left, the page structure is visible with a 'Search' component containing facets for P9_SEARCH, P9_COLOR_ID, P9_DEPARTMENT_ID, P9_CLOTHING_ID, and P9_UNIT_PRICE. In the center, a 'Search Results' region is displayed with a 'Sort Order' of P9_ORDER_BY. The right side shows the 'Region' and 'Attributes' tabs for the 'Search Results' region. Under the 'SQL Query' tab, the following SQL code is shown:

```

SELECT "PRODUCT_ID",
       "PRODUCT_NAME",
       "UNIT_PRICE",
       "PRODUCT_DETAILS",
       "PRODUCT_IMAGE",
       "IMAGE_TYPE",
       "IMAGE_TYPE",
       "IMAGE_CHARSET",
       "IMAGE_LAST_UPDATED",
       "COLOR_ID",
       (

```

Under Appearance section:

- Click **Template Options**. For Style - select **Style A**.

The screenshot shows the 'Template Options' dialog for the 'Search Results' region. Under the 'Style' tab, the 'Style A' dropdown is highlighted with a red box. Other options like 'Default', 'Style B', and 'Style C' are also visible. The 'OK' button at the bottom right is also highlighted with a red box.

- Click OK

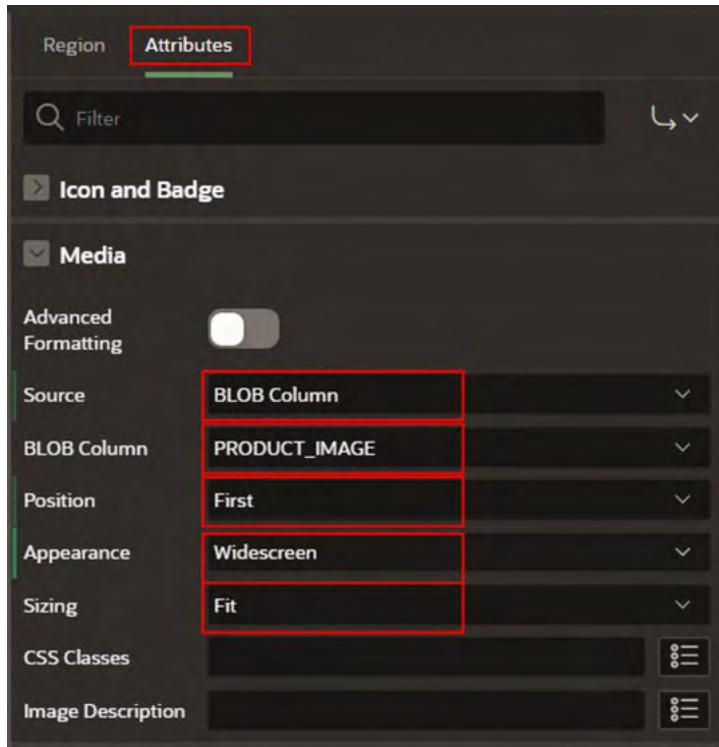
2. Click **Attributes** and apply the following changes:

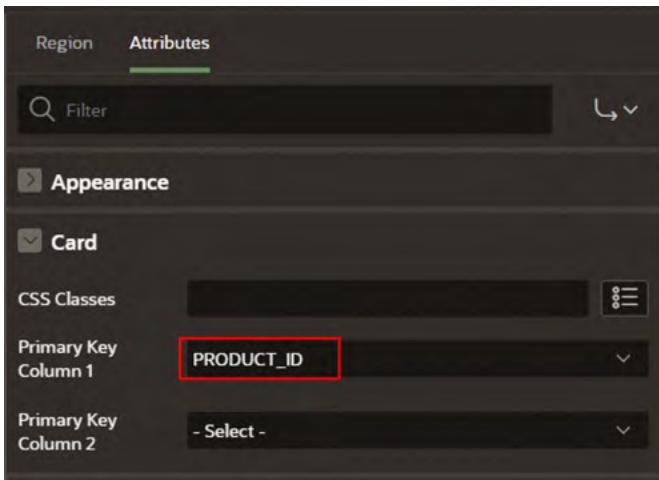
- Click **Attributes** and apply the following changes:
 - For Layout - select **Grid**.
 - For Grid Columns - select **Auto**.
 - Under Title section:
 - For Column - select **PRODUCT_NAME**.
 - Under Subtitle section:
 - Set Advanced Formatting to On.

- For HTML Expression - enter the following:

```
<small>&BRAND.</small><br />
<b class="u-success-text u-pullRight" id="message_&PRODUCT_ID.">
{if QUANTITY/} &QUANTITY. in cart {endif/}
</b>
<b>$&UNIT_PRICE.</b>
```

- Under Media section:
 - For Source - select **BLOB Column**.
 - For BLOB Column - select **PRODUCT_IMAGE**.
 - For Position - select **First**.
 - For Appearance - select **Widescreen**.
 - For Sizing - select **Fit**.
- Under Card section:
 - For Primary Key Column 1 - select **PRODUCT_ID**
- Click **Save**.

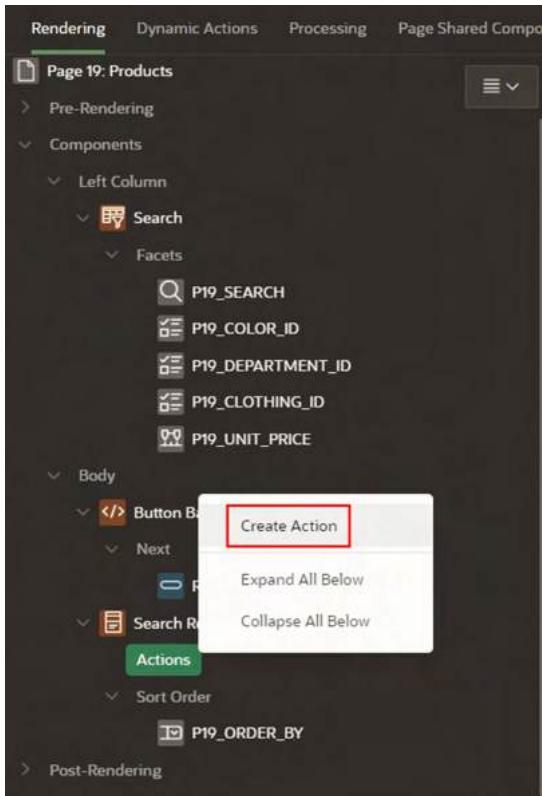




Create Actions

Customers need a way to shop the products, so in this task you will add an action to allow customers to learn more about the product.

1. Navigate to **Search Results** (left pane).
2. On Actions, right-click **Create Action**.

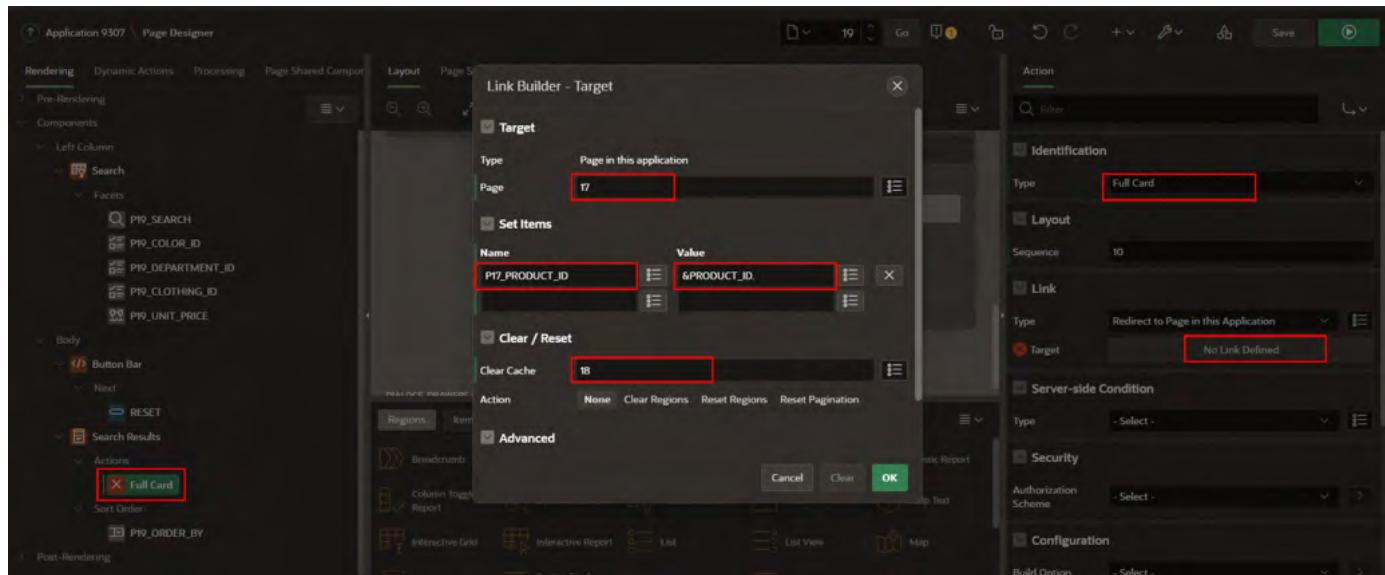


3. In the Property Editor (right pane), enter the following:
- For Type - select **Full Card**.
 - For Target - click **No Link Defined** and do the following:
 - For Page - enter **17**.
 - For Set Items, enter:

Table 1: Manage Cards, Faceted Search, and Smart Filters | Lab 2: Improve and Customize the Faceted Search and Cards regions

| Name | Value |
|----------------|--------------|
| P17_PRODUCT_ID | &PRODUCT_ID. |

- For Clear Cache, enter **18**.
- Click **OK**.

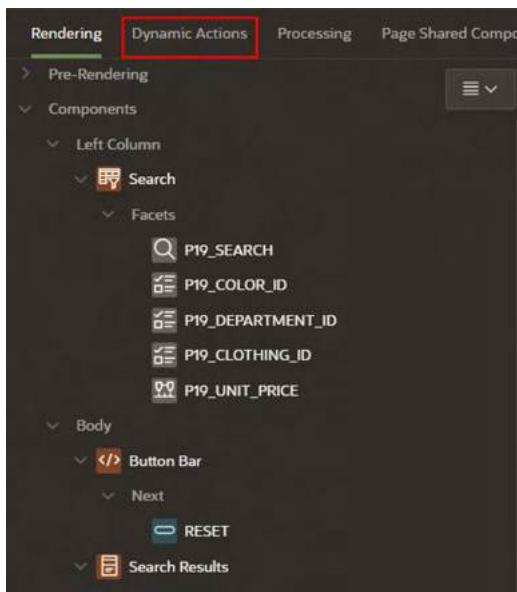


Add Dynamic Actions

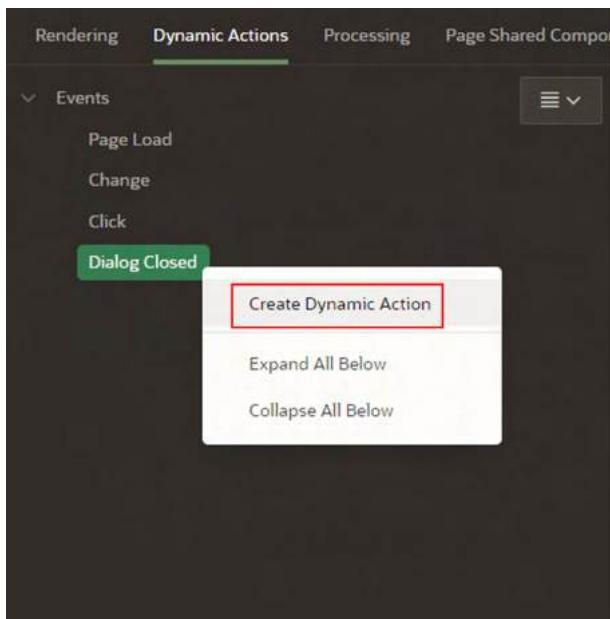
In this task, you will create two dynamic actions:

- To show a success message when a product is added/edited/removed from the shopping cart
- To update the badge and icon shown in the navigation bar after the customer has added/edited/removed a product from the shopping cart

1. Navigate to **Dynamic Actions** tab (left pane).

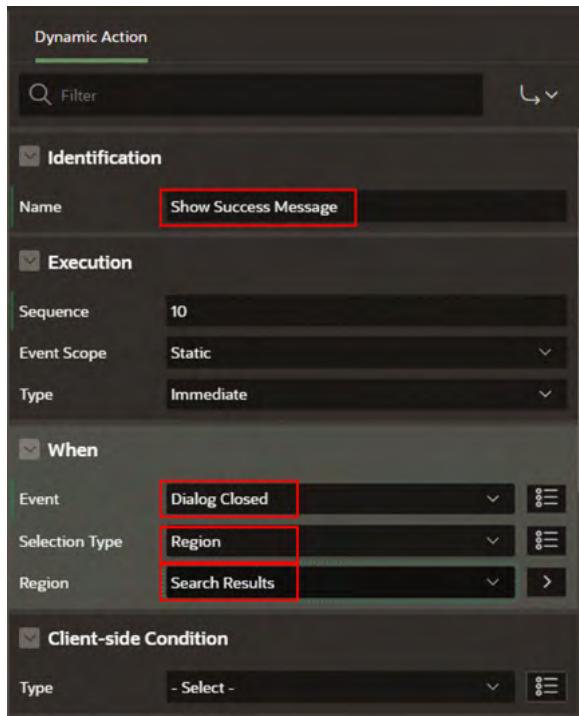


2. Right-click **Dialog Closed** and select **Create Dynamic Action**.



3. In the Property Editor, enter the following:
 - Under the Identification section:
 - For Name - enter Show Success Message.
 - Under When section:
 - For Event - select **Dialog Closed**.
 - For Selection Type - select **Region**.

- For Region - select **Search Results**.

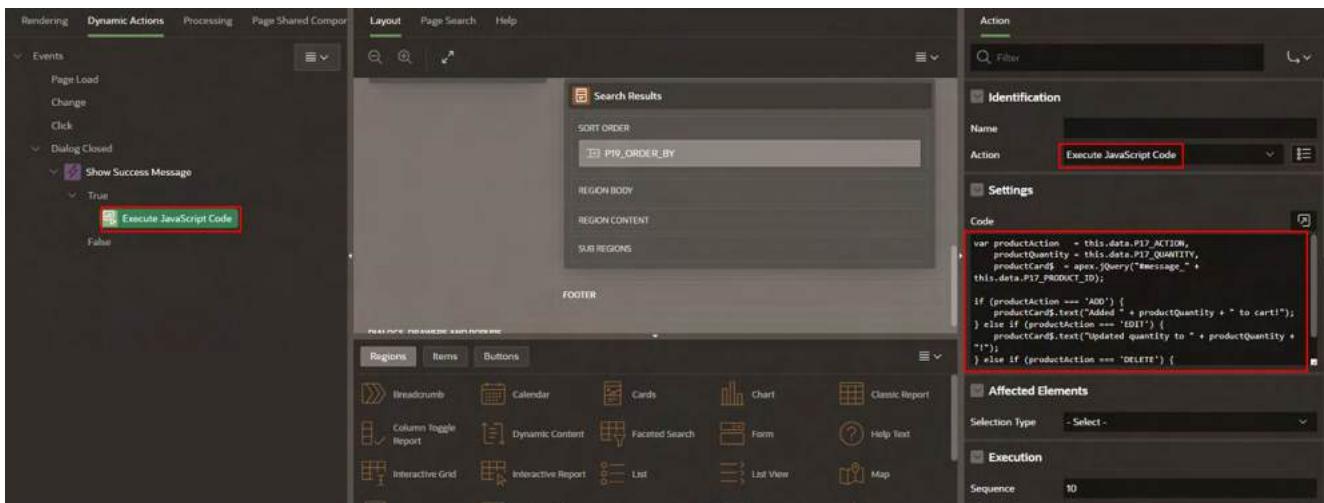


4. Navigate to **Refresh** Action.

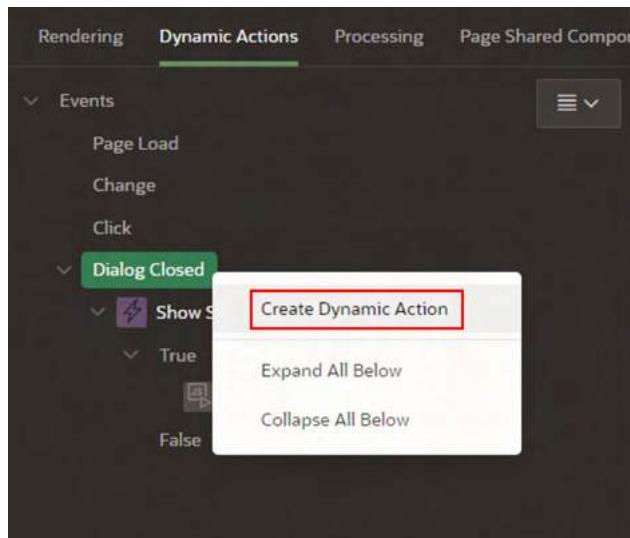
- Under Identification section:
 - For Action - select **Execute JavaScript Code**.
- Under Settings section:
 - For Code - enter the following JavaScript Code:

```
var productAction = this.data.P17_ACTION,
    productQuantity = this.data.P17_QUANTITY,
    productCard$ = apex.jQuery("#message_" +
this.data.P17_PRODUCT_ID);

if (productAction === 'ADD') {
    productCard$.text("Added " + productQuantity + " to cart!");
} else if (productAction === 'EDIT') {
    productCard$.text("Updated quantity to " + productQuantity +
"!");
} else if (productAction === 'DELETE') {
    productCard$.text("Removed from cart!");
}
```



5. Create a second dynamic action. Right-click **Dialog Closed** and select **Create Dynamic Action**.

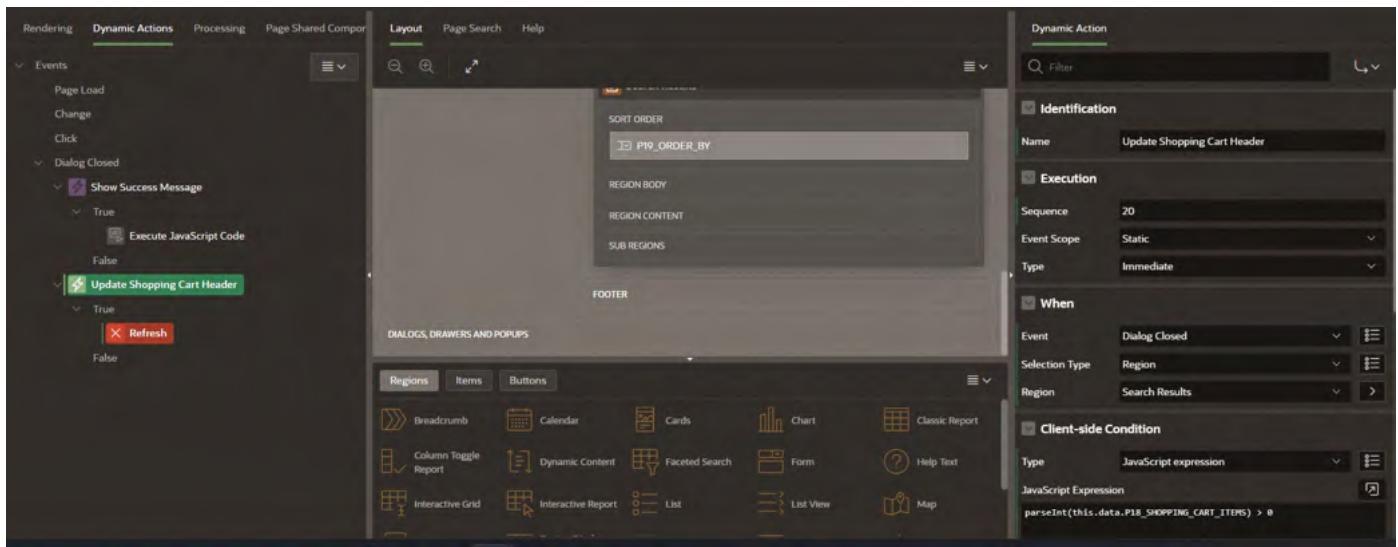


6. In the Property Editor, enter the following:

- Under Identification section:
 - For Name - enter **Update Shopping Cart Header**.
- Under When section:
 - For Event - select **Dialog Closed**.
 - For Selection Type - select **Region**.
 - For Region - select **Search Results**.
- Under Client-side Condition:
 - For Type - select **JavaScript expression**.

- For JavaScript Expression, enter the following:

```
parseInt(this.data.P18_SHOPPING_CART_ITEMS) > 0
```

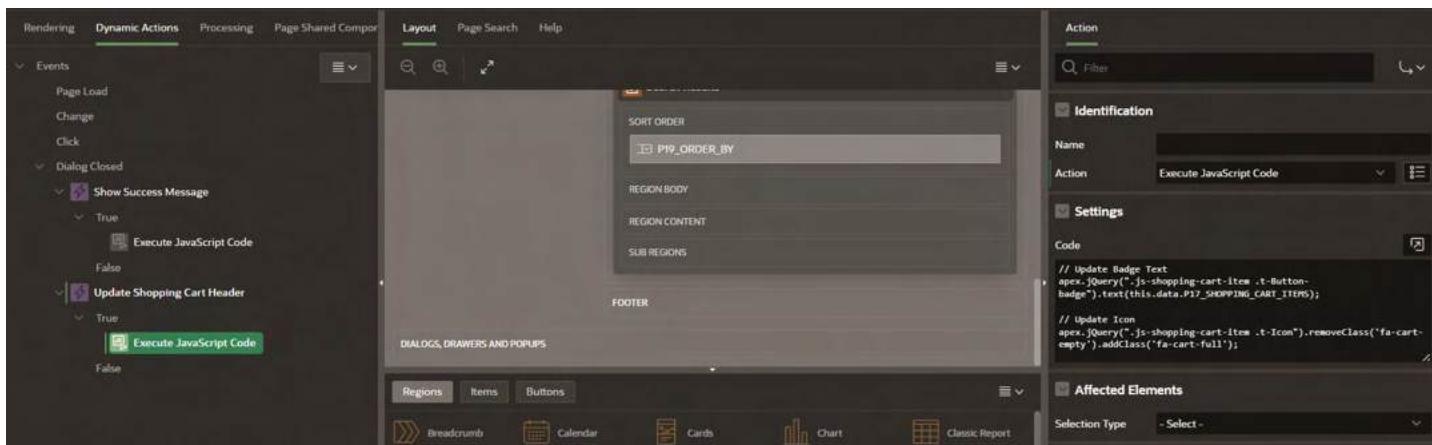


7. Navigate to the Refresh action.

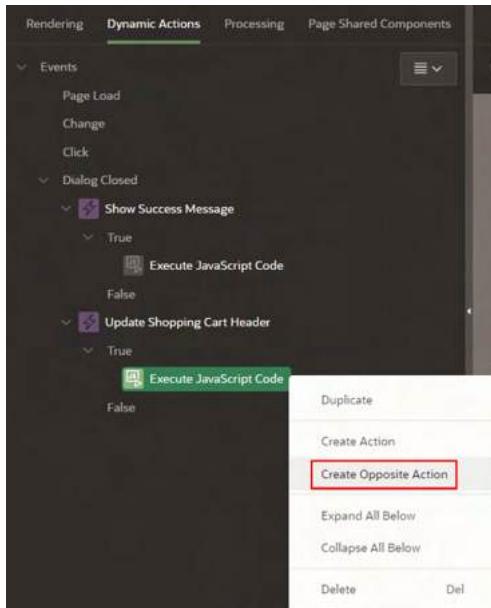
- Navigate to the **Refresh** action.
 - For Action - select Execute JavaScript Code.
 - Under Settings section:
 - For Code - enter the following JavaScript Code:

```
// Update Badge Text
apex.jQuery(".js-shopping-cart-item .t-Button-badge").text(this.data.P17_SHOPPING_CART_ITEMS);

// Update Icon
apex.jQuery(".js-shopping-cart-item .t-Icon").removeClass('fa-cart-empty').addClass('fa-cart-full');
```



8. Create an opposite action. In the Dynamic Actions tab (left pane), navigate to the newly dynamic action.
9. Right-click **Execute JavaScript Code** and select **Create Opposite Action**.

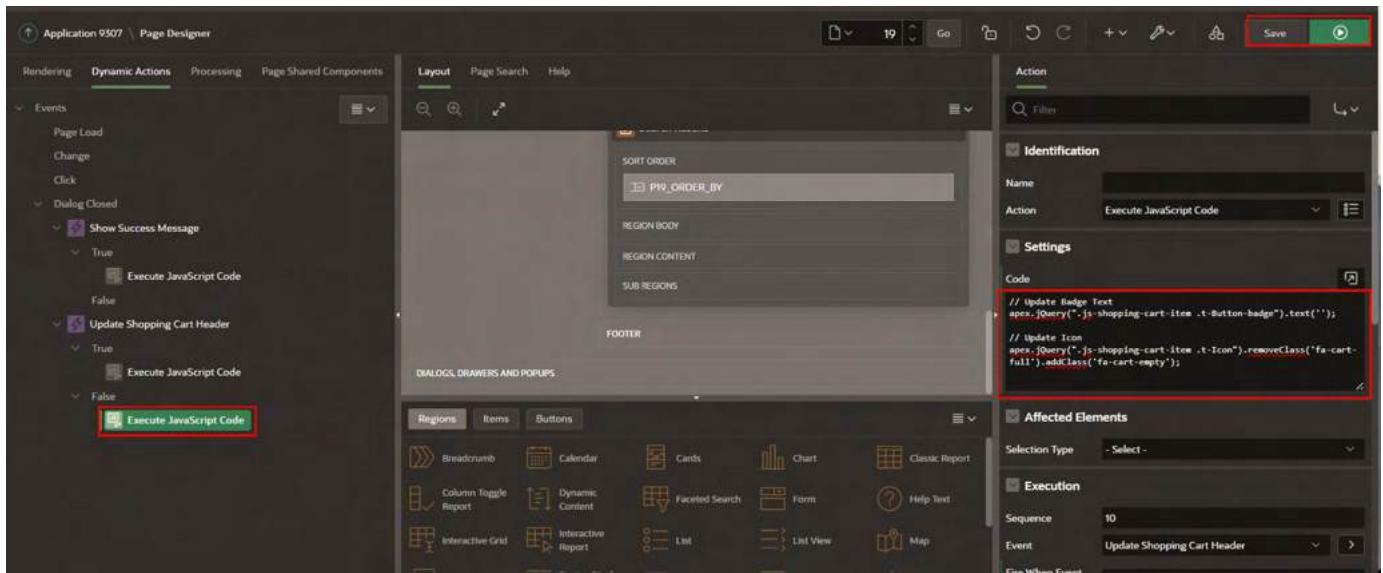


10. Navigate to the **Execute JavaScript Code** action.

- Under the Identification section:
 - For Action - select **Execute JavaScript Code**.
- Under the Settings section:
 - For Code - enter the following JavaScript Code:

```
// Update Badge Text
apex.jQuery(".js-shopping-cart-item .t-Button-badge").text('');

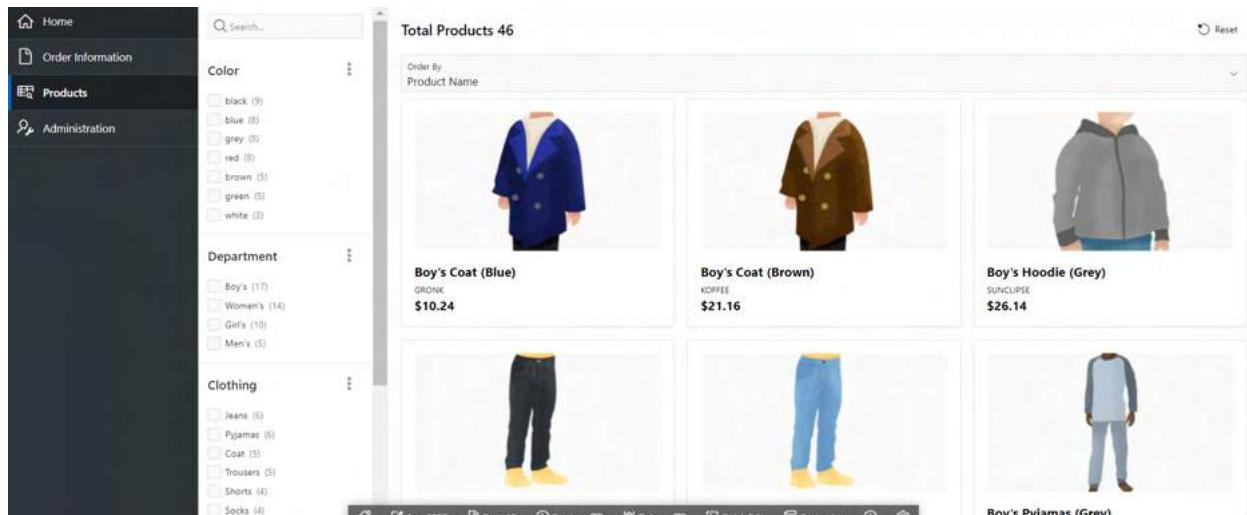
// Update Icon
apex.jQuery(".js-shopping-cart-item .t-Icon").removeClass('fa-cart-full').addClass('fa-cart-empty');
```



11. Click Save and Run Page.

Run the Products Page

1. After running products page, your page will look like the following image:



You now know how to enhance faceted search and cards region. You may now **proceed to the next lab.**

Practice: Creating and Using Forms

Practice 1: Create and Customize Forms

Overview

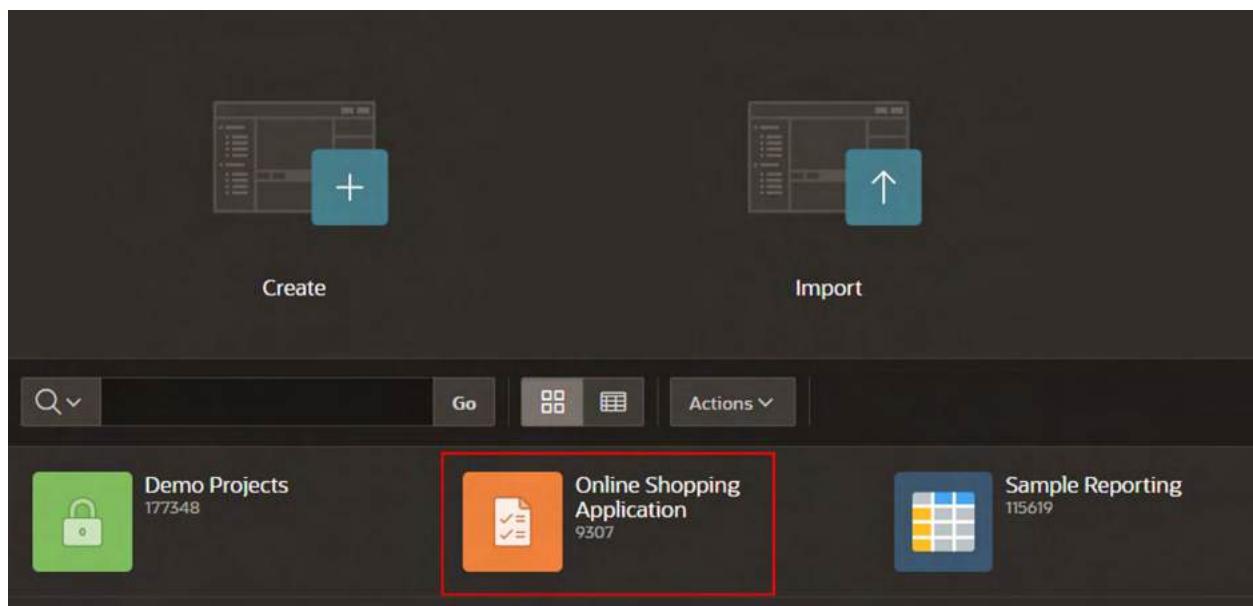
In this lab, you will learn how to create and customize a Form and then you will link the form to an Interactive Report.

Downloads

- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SOL Workshop](#) workshops.

Create a Product Details Form

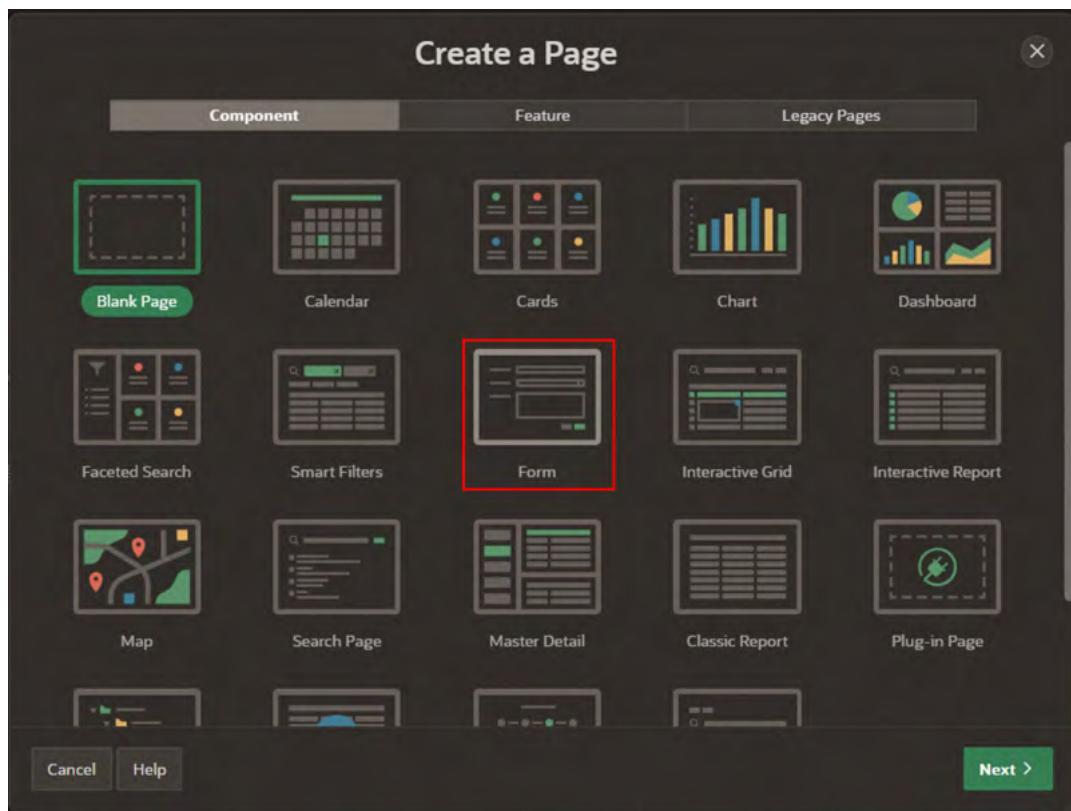
1. Navigate to the **App Builder**. Then, click **Online Shopping Application**.



2. Now, click Create Page.



3. For Create a Page: Select Page Type - select **Component** and then select **Form**.



4. For Page Attributes, enter the following:

Under Page Definition:

- For Page Number, enter 21.
- For Page Name, enter **Product Details**.

- For Page Mode, select **Modal Dialog**.

Under Data Source

- For **Table/View Name**, Select **PRODUCTS**. Click **Next**.

Create Form

Page Définition

- * Page Number: 21
- * Name: Product Details
- Page Mode: **Modal Dialog**

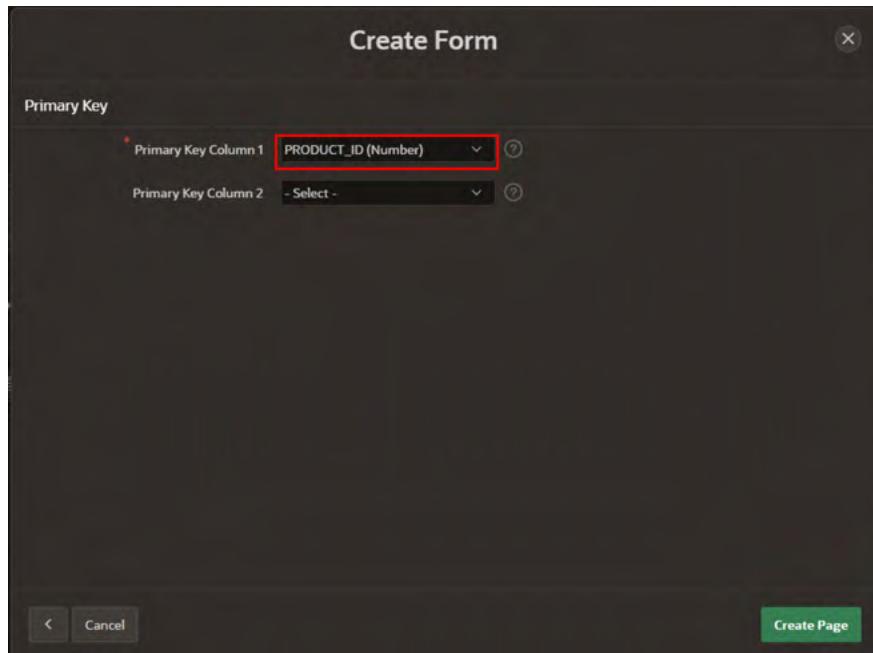
Data Source

- Data Source: **Local Database**
- Source Type: **Table**
- * Table / View Owner: WKSP_APEXHANDSONLAB22
- * Table / View Name: **PRODUCTS**

Navigation

< Cancel **Next >**

5. Under Primary key, for Primary Key Column1 - Ensure the **PRODUCT_ID** is selected as the primary key. Then click **Create Page**.



Link a Report to a Form

1. Navigate to the **App Builder**. Then, click **Online Shopping Application** and run **Application**. From the runtime application, navigate to the **Manage Products** page in **Page Designer**.

Given that you run this app from the APEX App Builder, you will find the Developer Toolbar at the bottom of the screen. *{Note: End users who log directly into the app will not see this toolbar.}*

In the Developer Toolbar, click **Page 14**.

| User Role | Count |
|---------------|-------|
| Administrator | 1 |
| Contributor | 0 |
| Reader | 0 |

Manage Products

| Product Name | Unit Price | Product Details | Product Image | Color | Department | Clothing |
|------------------------|------------|--------------------------|--------------------------|-------|------------|----------|
| Boy's Coat (Blue) | 10.24 | Download | Download | brown | Boy's | Coat |
| Boy's Coat (Brown) | 21.16 | Download | Download | brown | Boy's | Coat |
| Boy's Hoodie (Grey) | 26.14 | Download | Download | grey | Boy's | Hoodie |
| Boy's Jeans (Black) | 16.64 | Download | Download | black | Boy's | Jeans |
| Boy's Jeans (Blue) | 22.98 | Download | Download | blue | Boy's | Jeans |
| Boy's Pyjamas (Grey) | 23.32 | Download | Download | grey | Boy's | Pyjamas |
| Boy's Shirt (Black) | 37.34 | Download | Download | black | Boy's | Shirt |
| Boy's Shirt (White) | 29.55 | Download | Download | white | Boy's | Shirt |
| Boy's Shorts (Blue) | 10.48 | Download | Download | blue | Boy's | Shorts |
| Boy's Socks (Black) | 19.58 | Download | Download | black | Boy's | Socks |
| Boy's Socks (Grey) | 19.16 | Download | Download | grey | Boy's | Socks |
| Boy's Socks (White) | 12.64 | Download | Download | grey | Boy's | Socks |
| Boy's Sweater (Green) | 44.17 | Download | Download | green | Boy's | Sweater |
| Boy's Sweater (Red) | 9.80 | Download | Download | red | Boy's | Sweater |
| Boy's Trousers (Black) | 39.32 | Download | Download | blue | Boy's | Trousers |
| Boy's Trousers (Blue) | 34.06 | Download | Download | blue | Boy's | Trousers |

2. In the Rendering tree (left pane), navigate to **Products**.

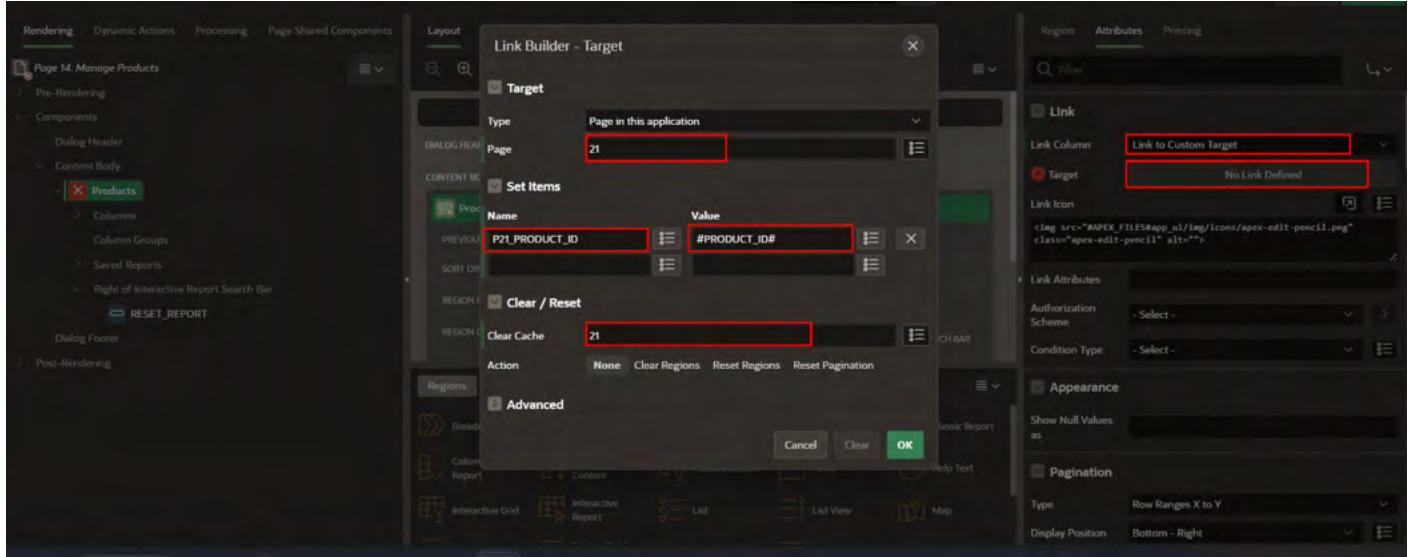
3. In the Property Editor (right pane), click **Attributes** and do the following:

- Under **Link**:
 - For Link Column - select **Link to a Custom Target**
 - Click **No Link Defined** Button and do the following in the popup:
 - For Page - enter 21.
 - For Set Items, enter:

| Name | Value |
|----------------|--------------|
| P21_PRODUCT_ID | #PRODUCT_ID# |

- For Clear Cache, enter 21.
- Click **OK**.

Click **Save**.



4. Click **Save** and then reload the **Manage Products** page in the runtime environment.
5. Now, In the **Manage Products** page, click **Pencil Icon** on the first row. Verify the **Product Details** model dialog page.

Manage Products

Go Actions ▾ Reset

| Product Name | Unit Price | Product Details | Product Image | Color | Department | Clothing |
|-----------------------|------------|-----------------|---------------|-------|------------|----------|
| Boy's Coat (Blue) | 10.24 | Download | Download | brown | Boy's | Coat |
| Boy's Coat (Brown) | 21.16 | Download | Download | brown | Boy's | Coat |
| Boy's Hoodie (Grey) | 26.14 | Download | Download | grey | Boy's | Hoodie |
| Boy's Jeans (Black) | 16.64 | Download | Download | black | Boy's | Jeans |
| Boy's Jeans (Blue) | 22.98 | Download | Download | blue | Boy's | Jeans |
| Boy's Pyjamas (Grey) | 23.32 | Download | Download | grey | Boy's | Pyjamas |
| Boy's Shirt (Black) | 37.34 | Download | Download | black | Boy's | Shirt |
| Boy's Shirt (White) | 29.55 | Download | Download | white | Boy's | Shirt |
| Boy's Shorts (Blue) | 10.48 | Download | Download | blue | Boy's | Shorts |
| Boy's Socks (Black) | 19.58 | Download | Download | black | Boy's | Socks |
| Boy's Socks (Grey) | 19.16 | Download | Download | grey | Boy's | Socks |
| Boy's Socks (White) | 12.64 | Download | Download | grey | Boy's | Socks |
| Boy's Sweater (Green) | 44.17 | Download | Download | green | Boy's | Sweater |
| Boy's Sweater (Red) | 9.80 | Download | Download | red | Boy's | Sweater |

Product Details

Product Name
Boy's Coat (Blue)

Unit Price
10.24

Product Details
Choose File
[Download](#)

Product Image
Choose File
[Download](#)

Image Mime Type

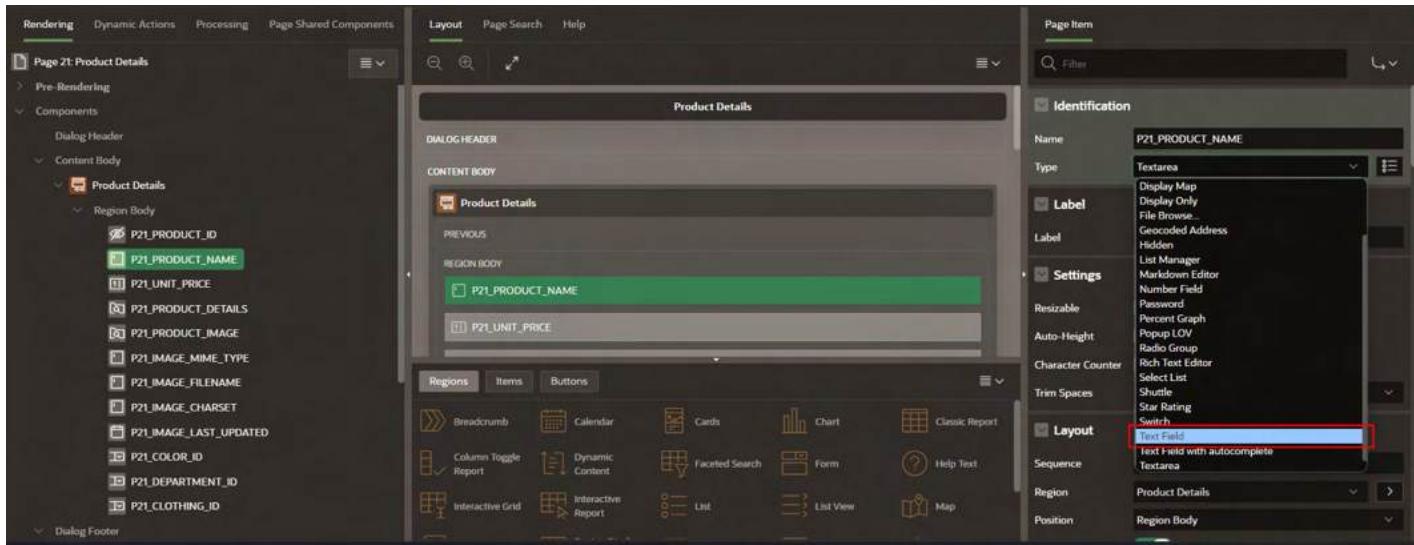
Image Filename

[Cancel](#) [Delete](#) [Apply Changes](#)

App 9307 Page 21 Session Debug Quick Edit Customize

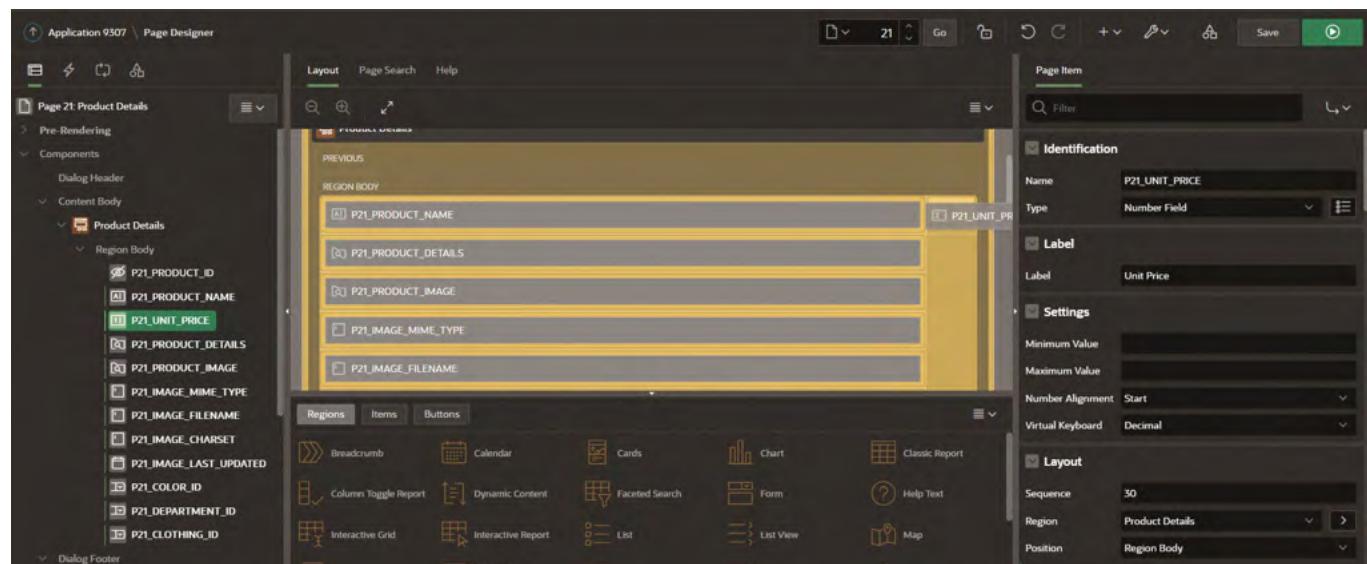
Enhance the Form

1. Navigate to **Page 21** in the **App Builder**. Then, click **P21_PRODUCT_NAME** in the Rendering Tree. Now, change the Identification Type to **Text Field** in the Property Editor.



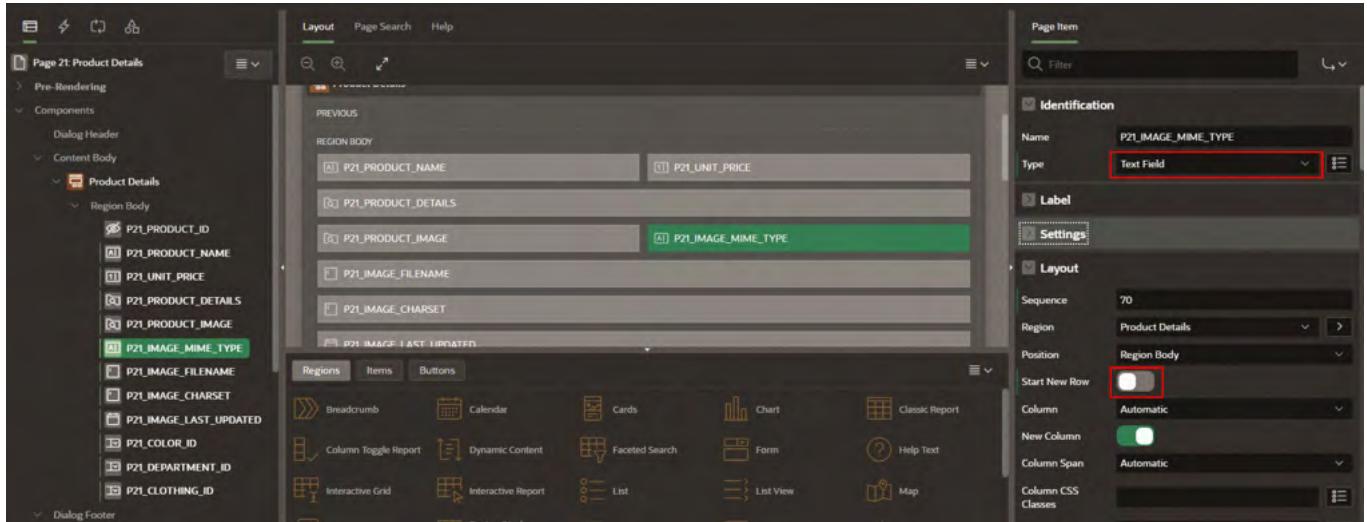
2. Items can readily be moved using drag-and-drop within Layout. If required, you can also drag new components such as regions, items, and buttons into the Layout pane from the Gallery, located directly below it.

In Page Designer, with **Page 21** loaded, within Layout (middle pane), click **P21_UNIT_PRICE** and continue to hold the mouse cursor down. Drag the item up and to the right, until it is directly after **P21_PRODUCT_NAME**, and a dark yellow box is displayed. Release the mouse cursor to drop the item in the new location.

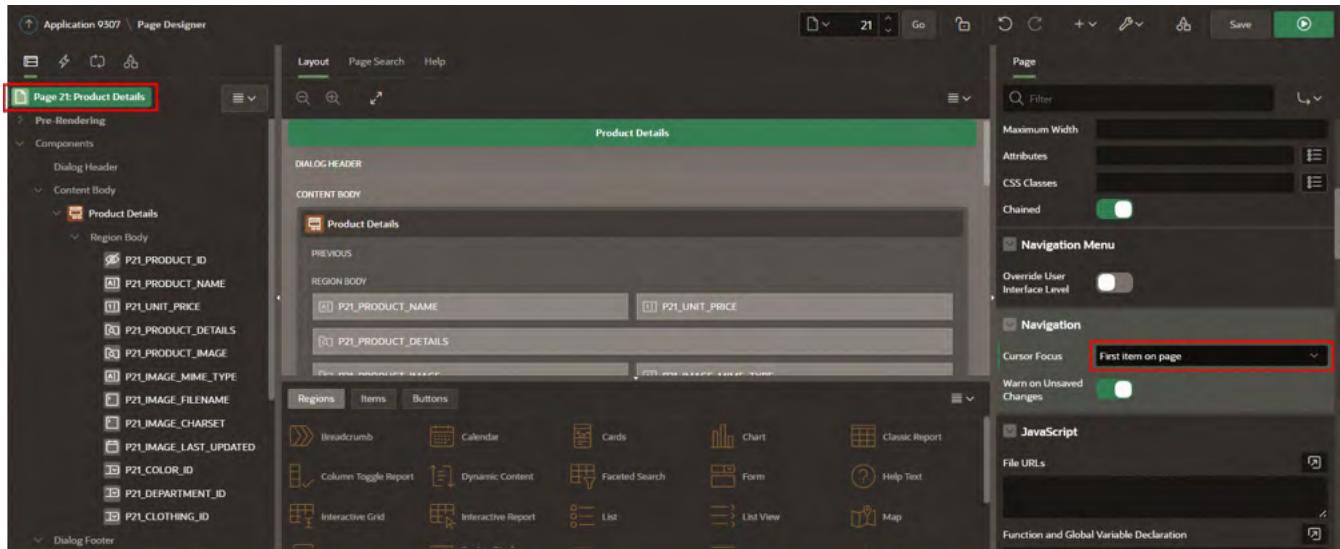


- As an alternative to using drag-and-drop you can also reposition items using attributes in the Property Editor.

In Page Designer, within Layout (or the Rendering tree in the left pane), select **P21_IMAGE_CHARSET**. In the Property Editor (right pane), under **Layout > Start New Row**.



- You now need to Focus on **First item on Page**. In **Page Rendering** (Left Pane), Select **Page 21: Product Details**. Then, In the **Property Editor**, Scroll down to Navigation and for **Cursor Focus**, select **First item on page**. Then, click **Save**.



- Now that you have customized the Form. You can reload **Manage Products** page in the runtime environment. Click **Edit** to view the Form Page.

The screenshot shows a product details form with the following fields:

- Product Name:** Boy's Coat (Blue)
- Unit Price:** 10.24
- Product Details:** Choose File, Download
- Product Image:** Choose File, Download
- Image Mime Type:** (empty)
- Image Filename:** (empty)
- Image Charset:** (empty)
- Image Last Updated:** (empty)

At the bottom, there are buttons for **Cancel**, **Delete**, and a blue **Apply Changes** button. The toolbar includes icons for file operations (New, Open, Save, Print), session management (Session, Debug, Quick Edit), and customization (Customize).

You now know how to create and customize a Form and then link the form to an Interactive Report. You may now **proceed to the next lab**.

Practice: Implementing Navigation in your Application

Practice 1: Improve Application

Overview

In this lab, you will learn how to make some pages publicly accessible.

In this lab, you will:

- Set the following pages as public pages:
 - Products
 - Shopping Cart
 - Order Information
- Disable the Navigation Menu
- Enhance the Navigation Bar

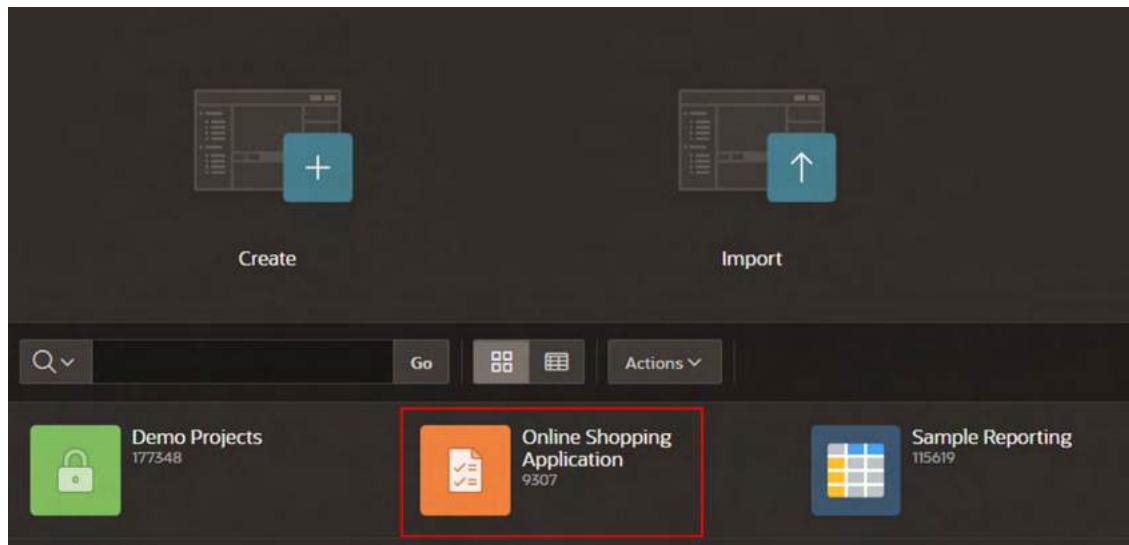
Downloads

- Did you miss out trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SOL Workshop](#) workshops.

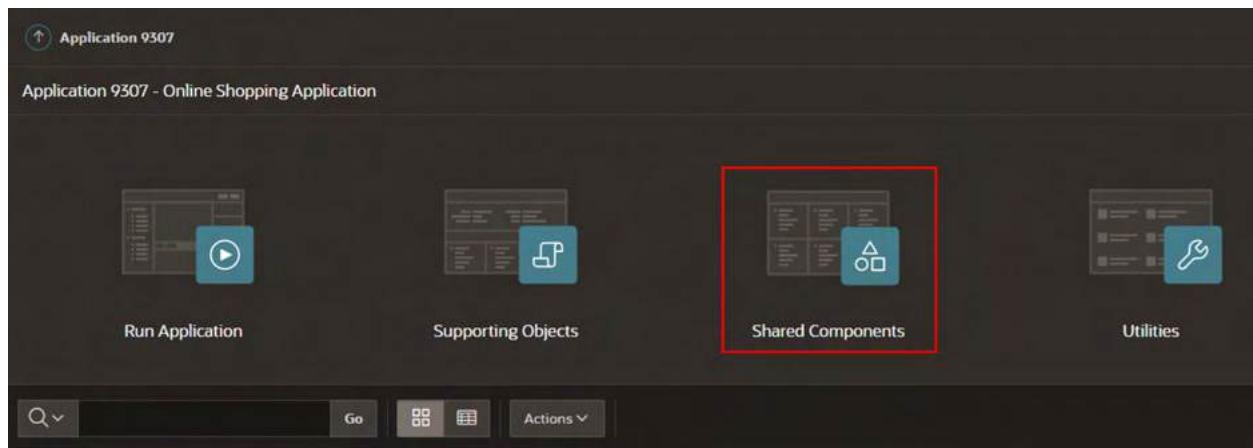
Set Products page as the Home Page

In this lesson, you will set the products page as the Home page and then you will remove the **Home** page under Navigation Menu.

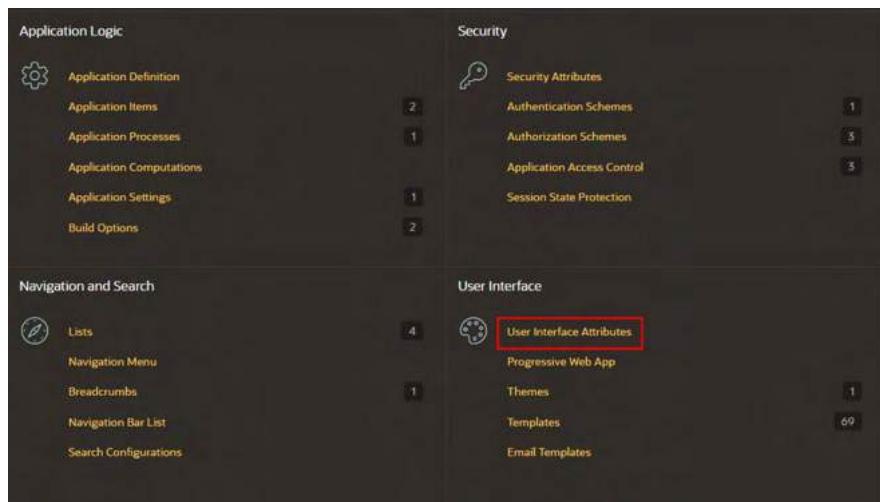
1. Navigate to the **App Builder**. Then, click **Online Shopping Application**.



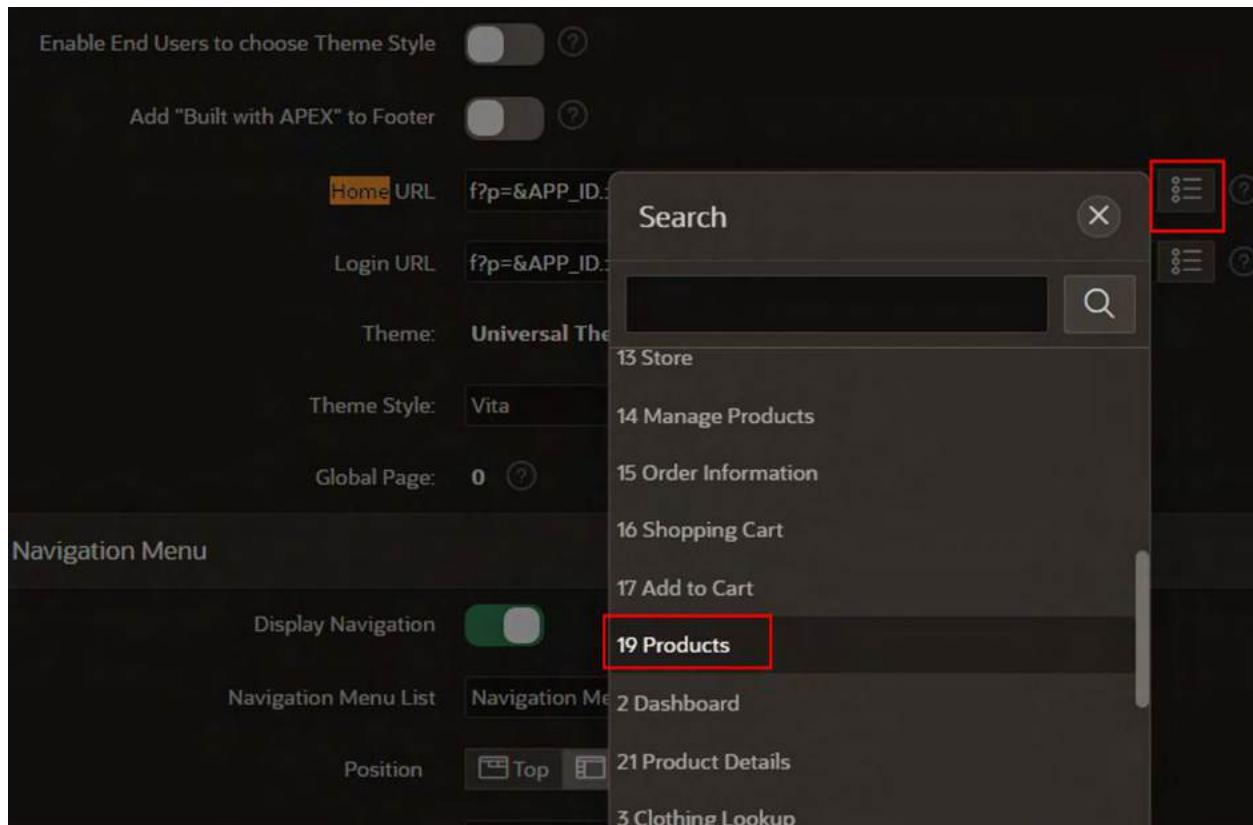
2. Now, select **Shared Components**.



3. Under User Interface, click **User Interface Attributes**.

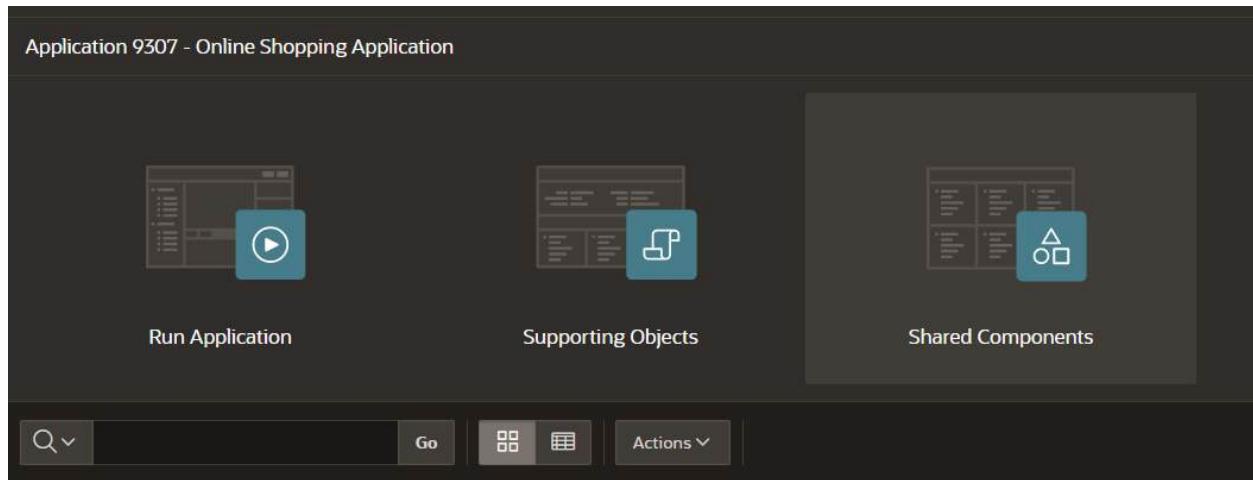


4. Now, under **Attributes**, select \equiv next to **Home URL**. Then, in the popup window, select **19 PRODUCTS** and click **Apply Changes**.



5. Because you have changed the **Home Page** to **Products** page, you now need to remove **Home** Page under Navigation Menu.
Navigate to **Shared Components**.

The screenshot shows the Oracle APEX 'Shared Components' page for Application 9307. The top navigation bar includes tabs for 'Definition', 'Security', 'Globalization', 'User Interface' (which is selected and highlighted in yellow), and 'Progressive Web App'. Below the tabs, the page title is 'Application 9307'. A horizontal menu bar at the top of the content area includes 'Show All' (selected), 'Icon', 'Logo', 'Attributes', 'Navigation Menu', 'Navigation Bar', 'JavaScript', 'CSS', 'Concatenated Files', and 'Advanced'. The 'Navigation Menu' section displays a list of menu items, with '19 Products' being one of them.



6. Under **Navigation**, Select **Navigation Menu**.

The screenshot shows the Oracle Application Builder navigation menu. It is divided into several sections:

- Application Logic**: Includes Application Definition, Application Items, Application Processes, Application Computations, Application Settings, and Build Options.
- Security**: Includes Security Attributes, Authentication Schemes, Authorization Schemes, Application Access Control, and Session State Protection.
- Navigation and Search**: Includes Lists, Navigation Menu (which is highlighted with a red box), Breadcrumbs, Navigation Bar List, and Search Configurations.
- User Interface**: Includes User Interface Attributes, Progressive Web App, Themes, Templates, and Email Templates.

7. Select **Navigation Menu**, then select **Pencil Icon** before **Home**.

| Name | Type | Entries | References | Updated | Navigation Bar |
|-----------------|--------|---------|------------|-------------|----------------|
| Navigation Menu | Static | 4 | 1 | 10 days ago | No |

| Sequence | Name | Target | Icon | Authorization Scheme | Build Option | Level | Parent Entry | Children | Conditional... | Upda |
|----------|-------------------|------------------------------------|------------------|-----------------------|--------------|-------|--------------|----------|----------------|-------|
| 10 | Home | !#p=&APP_ID:1&APP_SESSION:&DEBU... | fa-home | | | 1 | | | | 15 da |
| 20 | Order Information | !#p=&APP_ID:15&APP_SESSION:&DEB... | fa-file-o | | | 1 | | | | 11 da |
| 30 | Products | !#p=&APP_ID:19&APP_SESSION:&DEB... | fa-table-sear... | | | 1 | | | | 10 da |
| 10000 | Administration | !#p=&APP_ID:10000&APP_SESSION:&... | fa-user-wre... | Administration Rights | | 1 | | | | 15 da |

8. In the **List Entry** page, Click the **Delete** button and then confirm delete by clicking **OK**.

The 'List Entry' page displays the following details for the 'Home' entry:

- Entry:** List: Navigation Menu
- Parent List Entry:** - No Parent List Item -
- Sequence:** 10
- Image/Class:** fa-home
- Attributes:** (empty)
- Alt Attribute:** (empty)
- List Entry Label:** Home
- Target:** Target type: Page in this Application
- Page:** 1

Set Pages to Public

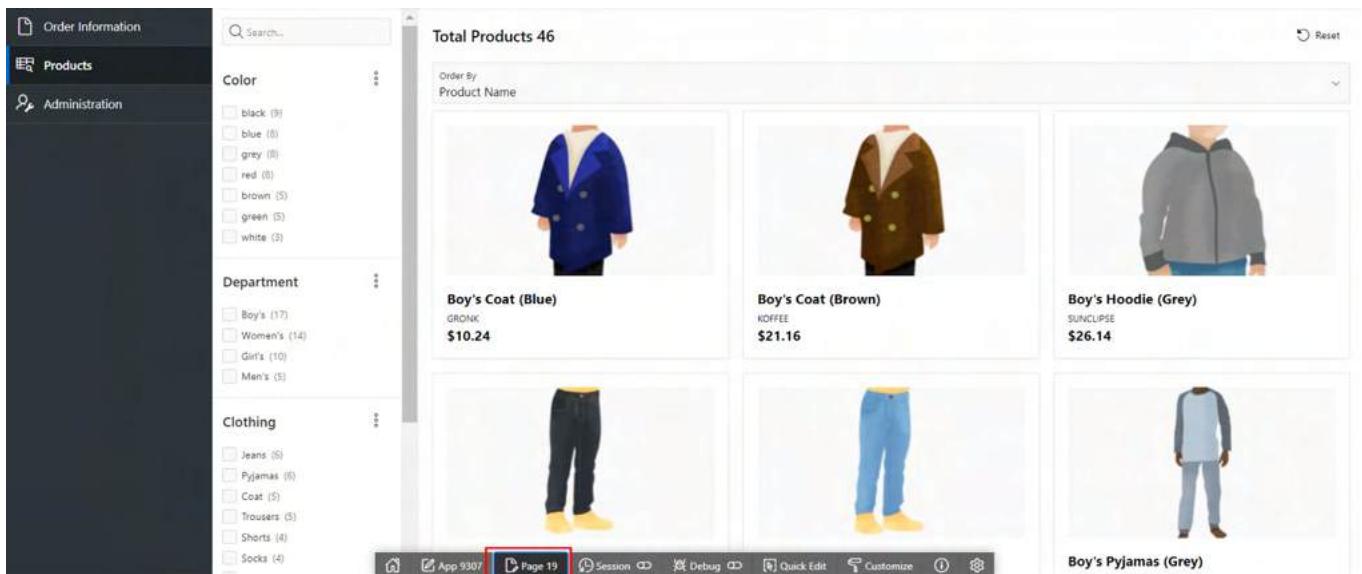
Your customers don't need to log in to the app to shop the products, so let's set the following pages as public:

- Page 19: Products
- Page 15: Order Information
- Page 16: Shopping Cart
- Page 17: Add to Cart

Follow these steps for the four listed pages:

1. From the runtime application, navigate to the **Products** page.

In the Developer Toolbar click **Edit Page 19**.



- Within Page Designer, in the Rendering tree (left pane), navigate to **Page 19: Products**.

In the Property Editor (right pane), apply the following change:

- Under Security section:
 - For Authentication - select **Page Is Public**.

- Click **Save**.

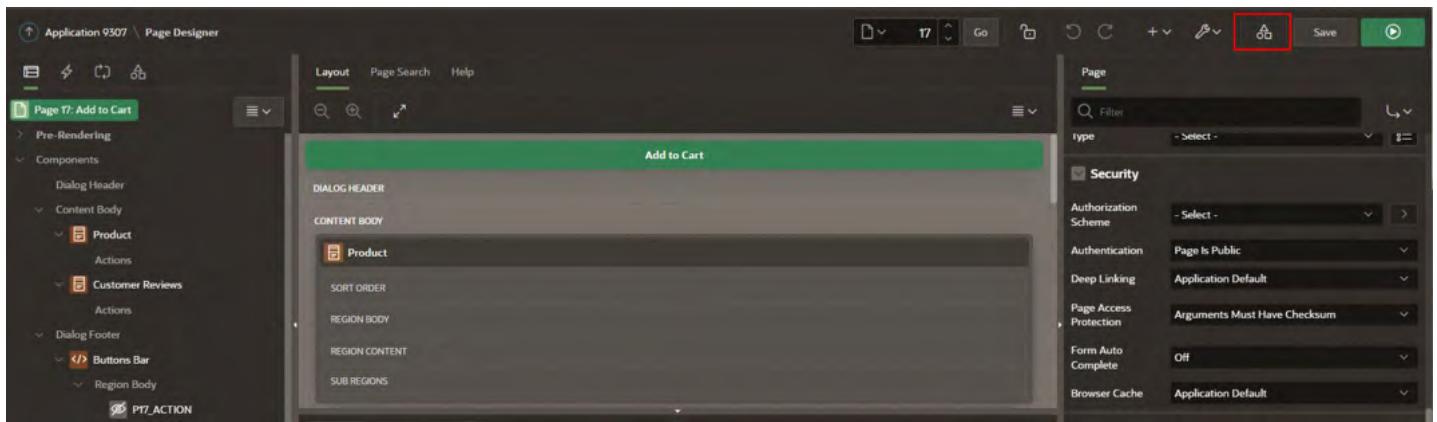
- Navigate to Page Finder and select **Page 15** from Dialog Page. Repeat steps 2-4.

- Repeat steps 2-4 for **Page 16: Shopping Cart** and **Page 17: Add to Cart**.

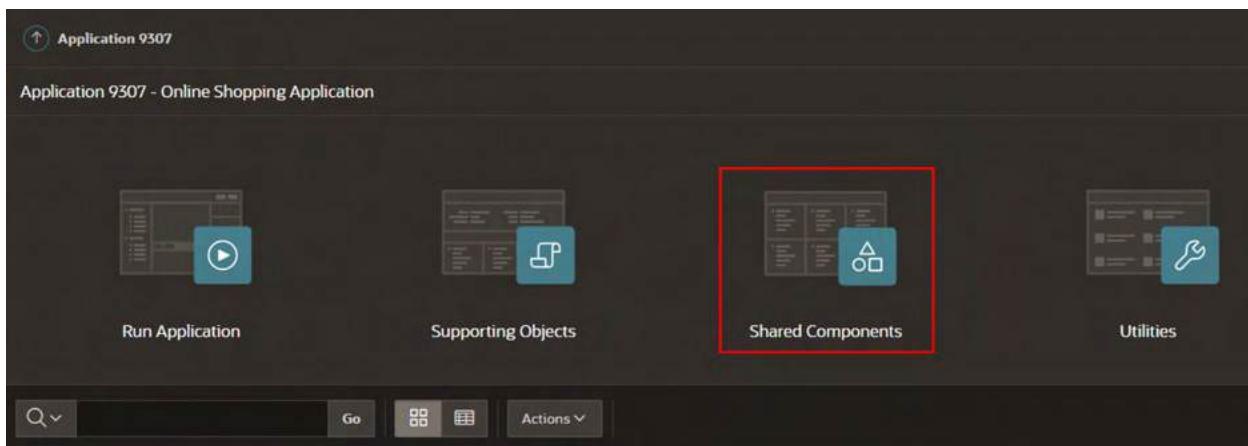
Remove the Navigation Menu

Because the home page is the Products page and this is a public access page, we do not need a navigation menu. In this task, you will turn off displaying the navigation menu.

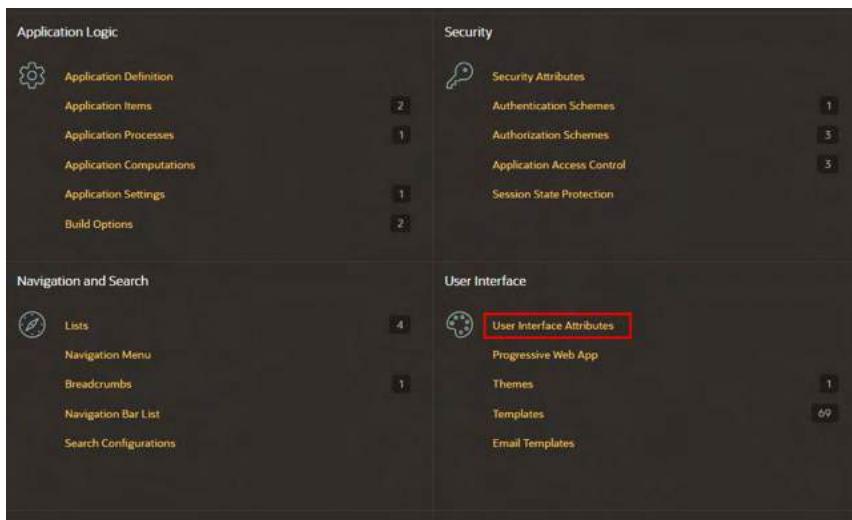
1. Within Page Designer, click the Shared Components icon at the top-right corner.



Alternatively, if you are in the App Builder, click Shared Components.

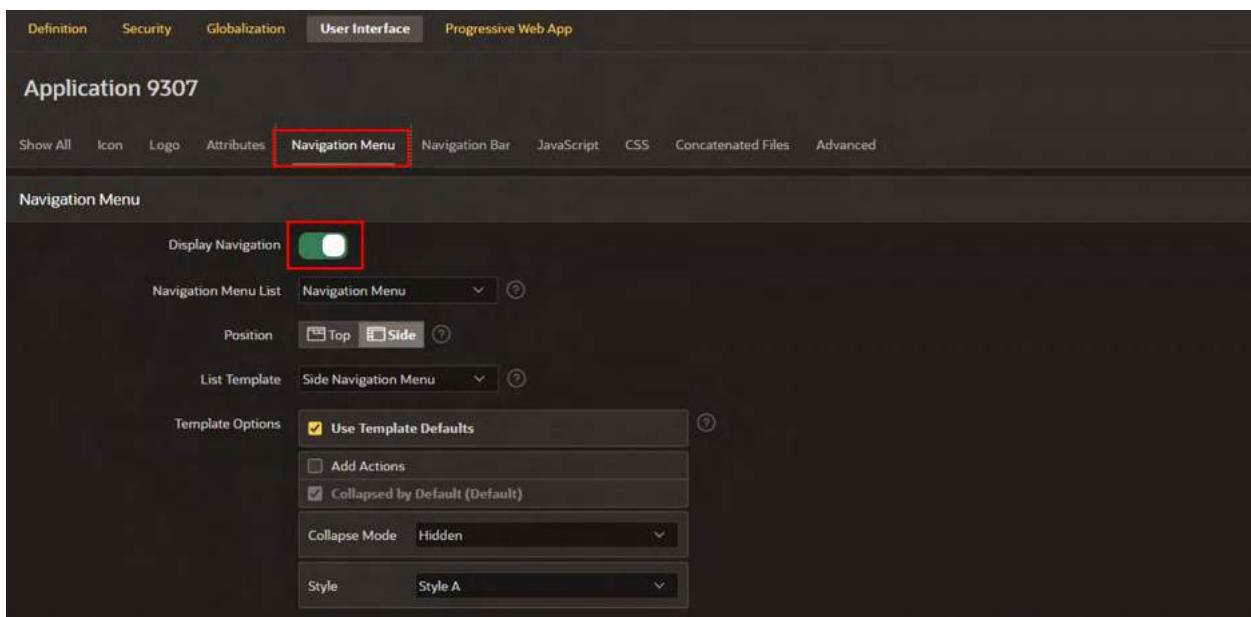


2. Under User Interface, click **User Interface Attributes**.



3. Click **Navigation Menu**.

4. Set Display Navigation to **Off**.



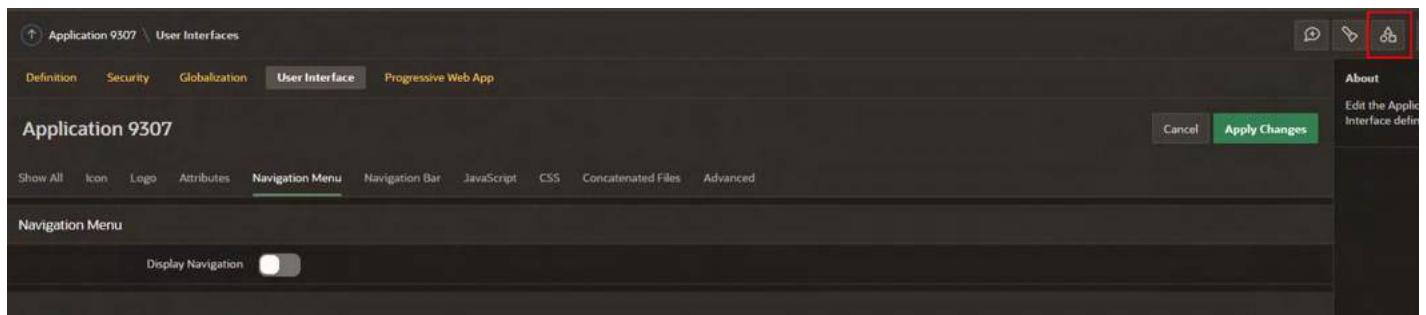
5. Click **Apply Changes**.

Enhance the Navigation Bar List

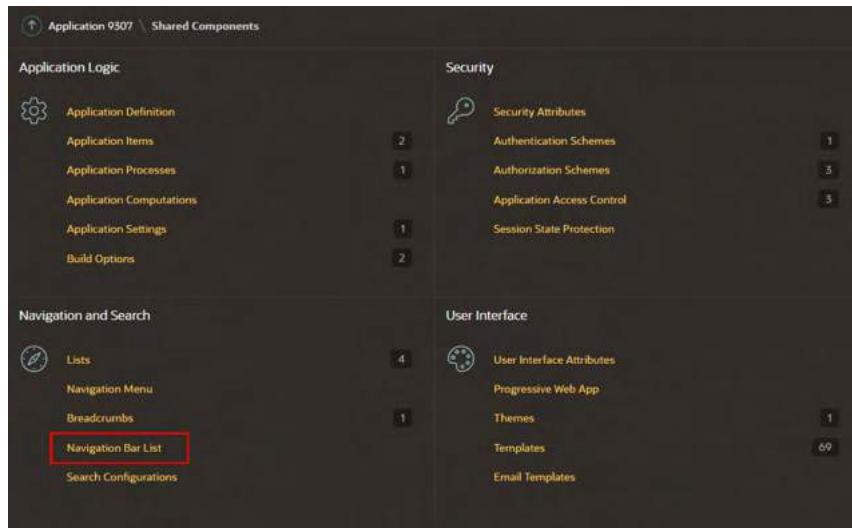
Add a new navigation bar entry to allow:

- Customers to go directly to the Shopping Cart
- Administrators to log in and access the administration page

1. Click the Shared Components icon at the top-right corner.



2. Under Navigation, click **Navigation Bar List**.



3. Click **Navigation Bar**.

| Name | Type | Entries | References | Updated | Navigation Bar | Navigation Menu | Subscribed From | Subscribers |
|----------------|--------|---------|------------|-------------|----------------|-----------------|-----------------|-------------|
| Navigation Bar | Static | 4 | 1 | 15 days ago | Yes | No | - | - |

4. Click **Create Entry**.

The screenshot shows the 'List: Navigation Bar' page in Oracle ADF. At the top, there are tabs for 'Show All', 'Name', 'List Entries', 'Subscription', 'Configuration', and 'Comments'. The 'Name' tab is selected. Below the tabs, there is a search bar with the placeholder 'Name' and the value 'Navigation Bar'. To the right of the search bar is a 'Create Entry >' button, which is highlighted with a red box. The main area is titled 'List Entries' and contains a table with four rows. The columns are 'Sequence', 'Name', 'Target', 'Icon', 'Authorization Scheme', 'Build Option', and 'Level'. The rows are: 1. Install App, #action\$pwa-install, fa-cloud-do..., -; 10. &APP_USER., #, fa-user, -; 20. ---, separator, -; 30. Sign Out, &LOGOUT_URL., fa-sign-out, -.

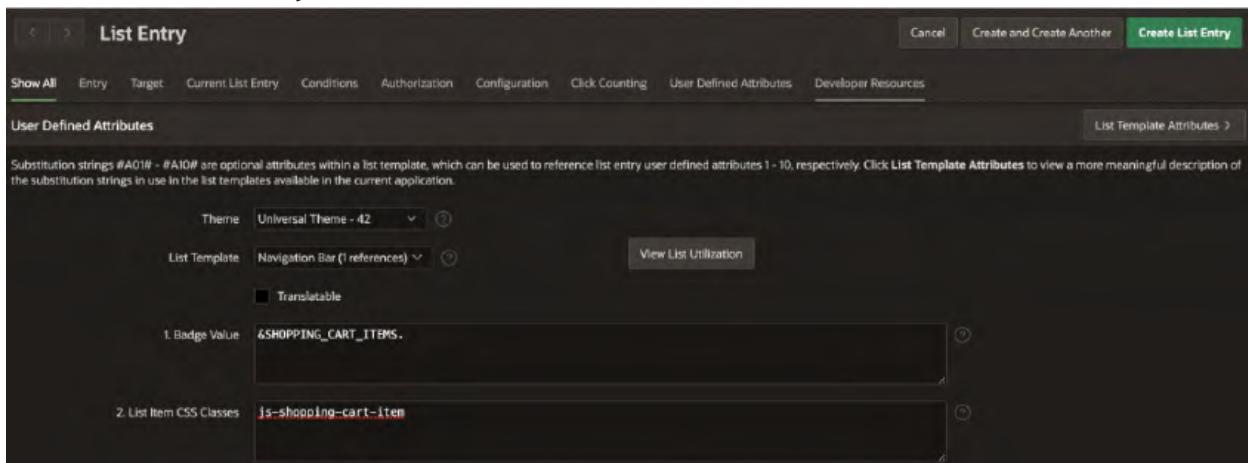
5. In the **List Entry** page, enter the following and click **Create List Entry**.

The screenshot shows the 'Entry' tab of the 'List Entry' page. The 'Sequence' field is set to 1. The 'Image/Class' field is set to &SHOPPING_CART_ICON. The 'List Entry Label' field is set to Shopping Cart. In the 'Target' section, the 'Page' field is set to 16. The 'Clear Cache' field is also set to 16. Other fields like 'Target type', 'Request', and 'Set these items' are visible but not highlighted.

- For Sequence - enter 1.
- For Image/Class – enter &SHOPPING_CART_ICON.
- For List Entry Label – enter Shopping Cart.
- For Page - select 16.
- For Clear cache - enter 16.

6. Scroll under **User Defined Attributes** and enter the following:
 - For 1. Badge Value - enter **&SHOPPING_CART_ITEMS.**
 - For 2. List Item CSS Classes - enter **js-shopping-cart-item.**

7. Click **Create List Entry.**



8. Click the **Pencil Icon** before **&APP_USER.**

| Sequence | Name | Target | Icon | Authorization Scheme | Build Option | Level | Parent Entry | Children | Conditional | Updated |
|----------|---------------|--------------------------------------|----------------|----------------------|--------------|-------|--------------|----------|-------------|---------|
| 1 | Shopping Cart | !\${&APP_ID}.16 &SESSION_&DEBUG_1... | &SHOPPING... | | | 1 | - | - | - | 2 sec |
| 10 | Install App | #action\$#a-pwa-install | fa-cloud-do... | | | 1 | - | - | - | 15 da |
| 20 | &APP_USER. | # | fa-user | | | 1 | - | 2 | - | 15 da |
| 30 | --- | separator | - | | | 2 | &APP_U... | - | - | 15 da |
| 40 | Sign Out | &LOGOUT_URL... | fa-sign-out | | | 2 | &APP_U... | - | - | 15 da |

- Under Authorization, for Authorization Scheme, select **Administration Rights**.
- Click **Apply Changes**.

The screenshot shows the 'List Entry' configuration interface. The 'Conditions' tab is active. In the 'Authorization' section, the 'Authorization Scheme' dropdown is set to 'Administration Rights' and is highlighted with a red box. Other tabs like 'Entry', 'Target', and 'Configuration' are also visible.

- Click **Create Entry** and enter the following:

| Name | Navigation Bar | Actions |
|----------------|------------------------------------|-----------------------|
| Shopping Cart | #p=&APP_ID:16&SESSION::&DEBUG_1... | fa-shopping-bag |
| Install App | #action\$#pwa-install | fa-cloud-download-alt |
| Administration | # | fa-user |
| separator | separator | - |
| Sign Out | &LOGOUT_URL | fa-sign-out |

- For Sequence - enter 5.
- For Image/Class - enter **fa-wrench**.
- For List Entry Label - enter **Administration**.
- For Page - select **10000**.

List Entry

Show All Entry Target Current List Entry Conditions Authorization Configuration Click Counting User Defined Attributes Developer Resources

Entry

List: Navigation Bar

Parent List Entry: - No Parent List Item -

Sequence: 5

Image/Class: fa-wrench

Attributes:

Alt Attribute:

List Entry Label: Administration

Target

Target type: Page in this Application

Page: 10000

Cancel Create and Create Another **Create List Entry**

12. Click **Create List Entry**.

You now know how to set page as public and manage Navigation bar and Navigation menu. You may now **proceed to the next lab**.

Practice: Using Themes and Themes Styles

Practice 1: Use Themes and Theme Styles

Overview

In this lab, you use the Theme Roller to save a new Theme Style. Then, you enable end users to apply this style while running the application.

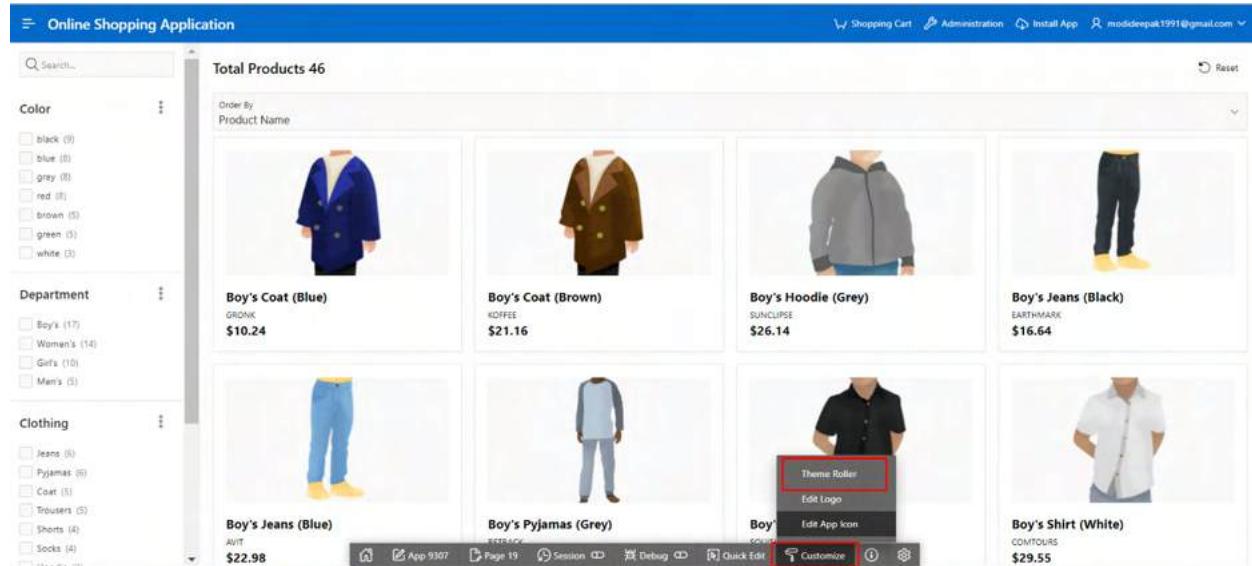
Downloads

- Did you miss out trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Use Themes and Theme Styles

In this hands-on lab, you use the Theme Roller to save a Theme Style. Then, you enable end users to apply this style while running the application.

1. Navigate to **App Builder** and then run the **Online Shopping Application**. In the Developer Toolbar, click **Customize** and then select **Theme Roller**.



2. The current Theme Style is **Vita**. Under **Global Colors**, click the color swatch to select new colors. To edit a specific component, expand a group and select new colors or styling of the component you wish to edit.

Modify the look by specifying colors of your choice and preview the changes. Once you are done, click **Save As**.

The screenshot shows the Oracle ADF Online Shopping Application. On the left, there are three filter panels: 'Color' (black, blue, grey, red, brown, green, white), 'Department' (Boy's, Women's, Girl's, Men's), and 'Clothing' (Jeans, Pyjamas, Coat, Trouser, Shorts, Socks). The main area displays a grid of products: 'Boy's Coat (Blue)' (\$10.24), 'Boy's Coat (Brown)' (\$21.16), 'Boy's Hoodie (Grey)' (\$26.14), 'Boy's Jeans (Blue)' (\$22.98), 'Boy's Pyjamas (Grey)', and 'Boy's Shirt (Black)'. On the right, a 'Theme Roller' dialog box is open, showing the 'Vita (Current)' theme selected under 'Select Theme'. The 'Global Colors' section is expanded, showing color swatches for Primary Accent (blue), Body Accent (light blue), Link Color (orange), and Focus Outline (orange). At the bottom of the Theme Roller are 'Save' and 'Save As' buttons.

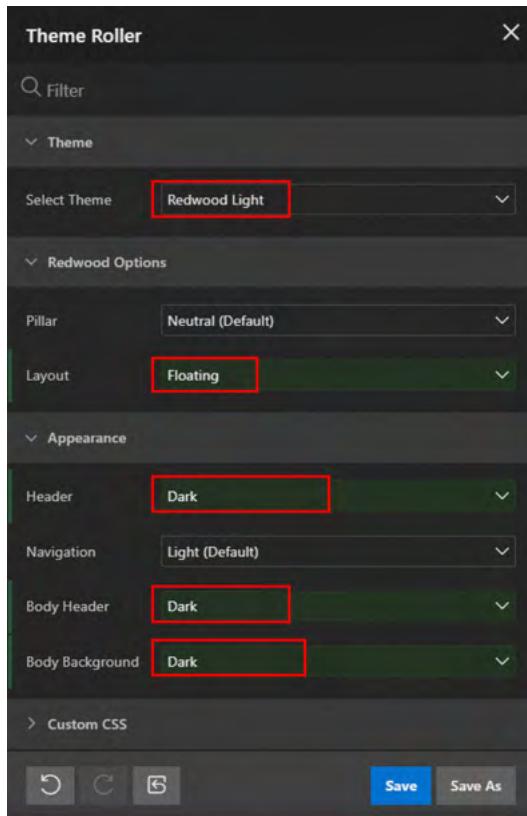
3. Under **Theme**, change the **Select Theme** to **Redwood Light**.

4. For Theme Attributes, enter the following:

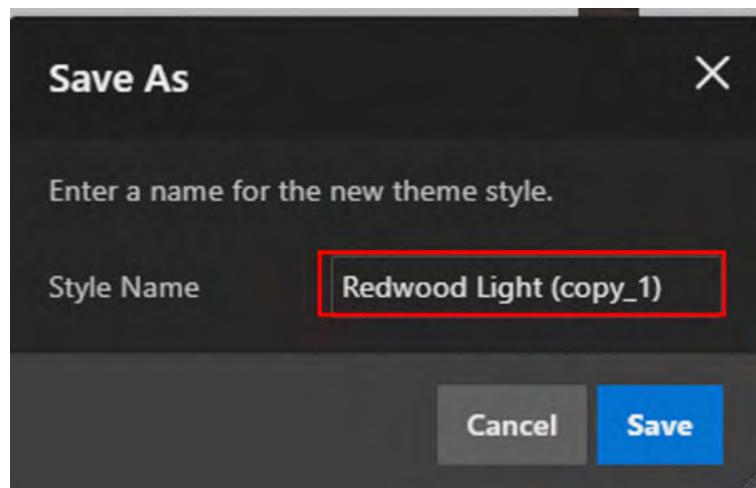
Under Redwood Options:

- For Layout, select **Floating**.
Under Appearance:
- For Header, select **Dark**.
- For Body Header, select **Dark**.
- For Body Background, select **Dark**.

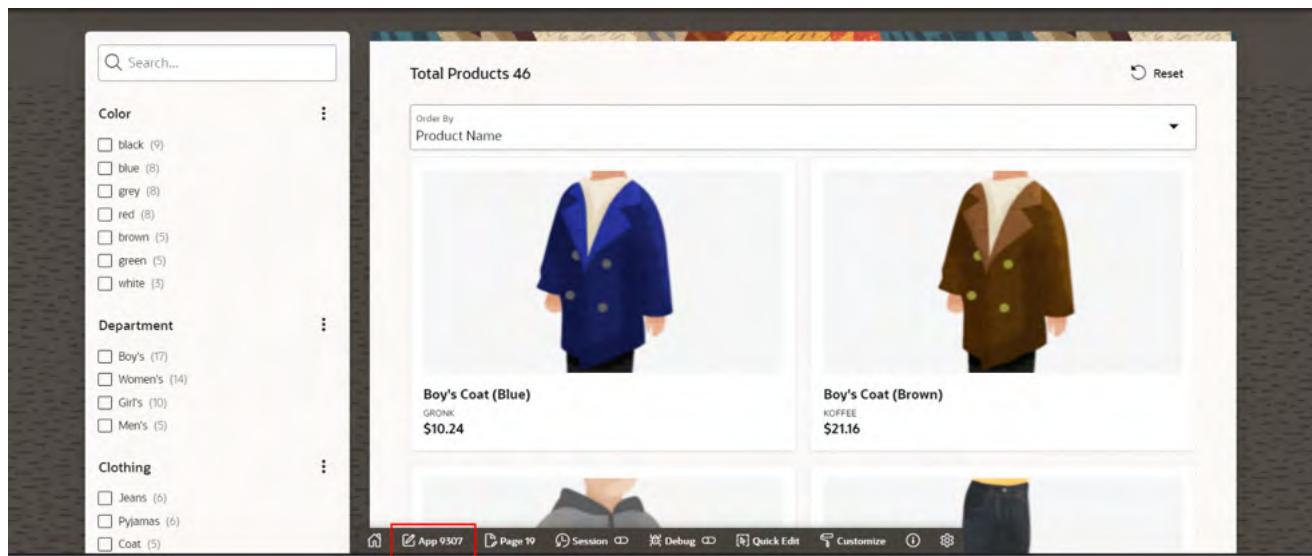
- Click **Save As**.



5. In the Save As dialog, enter **Redwood Light (Copy_1)** for Style Name and click **Save**.



6. Close the **Theme Roller** by clicking **X**. Click **App <n>** in the Developer Toolbar.



7. In the application home page, click **Shared Components**. Under User Interface, click **User Interface Attributes**.



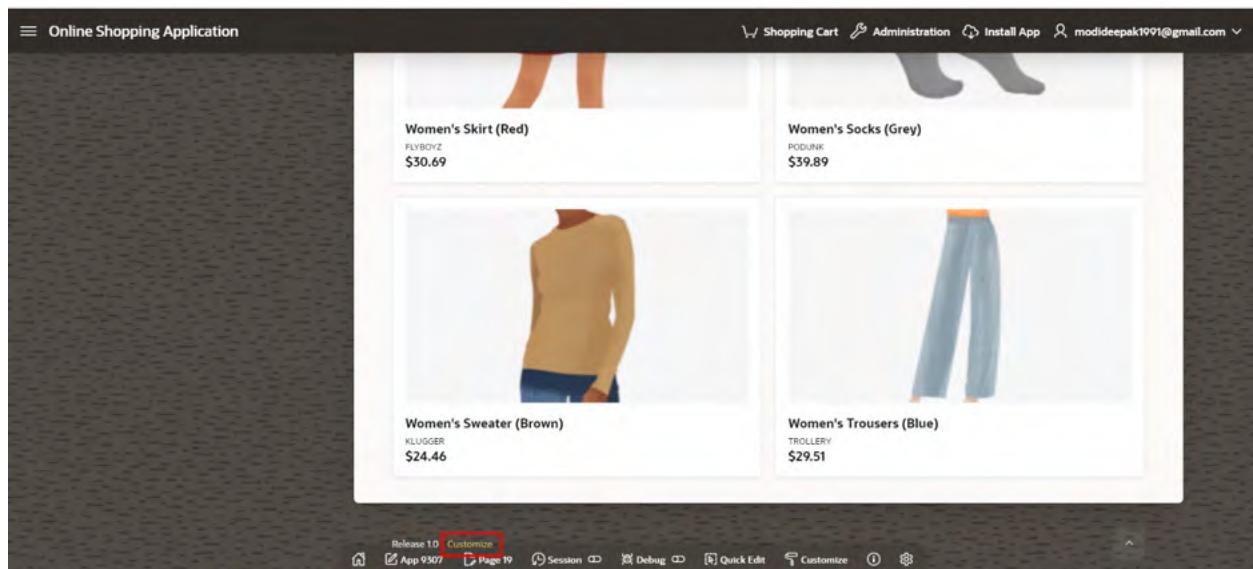
The screenshot shows the Oracle APEX Shared Components interface. It is organized into several sections:

- Application Logic** (left column):
 - Application Definition (2 items)
 - Application Items (1 item)
 - Application Processes (1 item)
 - Application Computations (1 item)
 - Application Settings (2 items)
- Security** (right column):
 - Security Attributes (1 item)
 - Authentication Schemes (1 item)
 - Authorization Schemes (3 items)
 - Application Access Control (3 items)
 - Session State Protection (1 item)
- Navigation and Search** (left column):
 - Lists (4 items)
 - Navigation Menu (1 item)
 - Breadcrumbs (1 item)
 - Navigation Bar List (1 item)
 - Search Configurations (1 item)
- User Interface** (right column):
 - User Interface Attributes (1 item)
 - Progressive Web App (1 item)
 - Themes (1 item)
 - Templates (66 items)
 - Email Templates (1 item)

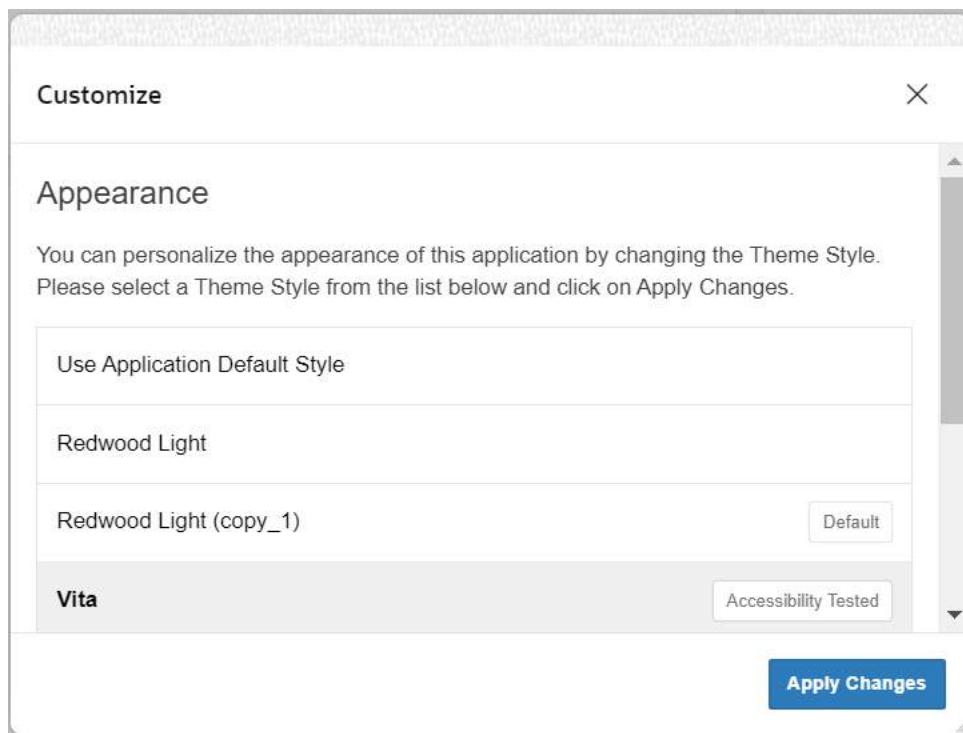
- Under **Attributes**, click **Enable End Users to Choose Theme Style**. Make sure this attribute is set to **On**. Notice that the new Theme Style you saved is displayed under Theme Style list. Click **Apply Changes**.

The screenshot shows the Oracle APEX User Interface Attributes page for Application 9307. The 'Attributes' tab is selected. The 'Enable End Users to choose Theme Style' checkbox is checked (highlighted with a red box). In the 'Theme Style' dropdown menu, 'Redwood Light (copy_1)' is selected (also highlighted with a red box).

9. Now, run the application. At the bottom of the page, click the **Customize** link.



10. Using the Customize dialog box, users can specify their Theme Style while running the application. Select the Theme Style that you saved in step 3 above and click Apply Changes.



11. The Theme Style is applied, and you see a message that says Preferences changed for user < username >.

The screenshot shows a shopping application interface. At the top, there's a blue header bar with the text "Online Shopping Application". Below it is a search bar and a message banner that says "Preferences changed for user". The main area has a title "Total Products 46" and a sorting dropdown set to "Order By Product Name". There are three filter panels on the left: "Color" (black, blue, grey, red, brown, green, white), "Department" (Boy's, Women's, Girl's, Men's), and "Clothing" (Jeans, Pyjamas, Coat, Trouser, Shorts, Socks). The main content area displays eight product items in a 2x4 grid:

- Boy's Coat (Blue)** by GROKK, \$10.24
- Boy's Coat (Brown)** by KOFFEE, \$21.16
- Boy's Hoodie (Grey)** by SUNCLIPSE, \$26.14
- Boy's Jeans (Black)** by EARTHMARK, \$16.64
- Boy's Jeans (Blue)** by AVIT, \$22.98
- Boy's Pyjamas (Grey)** by PISTACHE, \$19.99
- Boy's Shirt (Black)** by COFFEE, \$20.00
- Boy's Shirt (White)** by COMTOURS, \$29.55

At the bottom of the page, there are navigation links: Home, App 9307, Page 19, Session, Debug, Quick Edit, Customize, and Help.

You now know how to use the Theme Roller to save a new Theme Style. You may now **proceed to the next lab**.

Practice: Extending Application Capabilities

Practice 1: Manage Approvals Component

Overview

In this workshop, you will create an application that allows you to manage expenses using the approvals component. To explain the concepts and terms associated with Approval Management in APEX, we take the example of the Expense Tracker Application.

To apply for an expense, an employee logs in to the application and submits an expense request filling in the Type (Accommodation/Conference/Internet/ Miscellaneous Expenses) and Estimated Expenses. The task could be assigned to multiple potential owners, and then they can perform possible actions (Request information/Delegate/Release/Change Priority).

If the expense cost is more than 50000, then the Expense request, once approved by the immediate manager, will go to the next manager and so on, depending on the Expense amount. This is a typical use case for a multi-level approval.

Also, Deadlines and Expiration are set for a task. The potential owner will get an Email before 5 mins of task expiration. If the task is not approved or rejected before the expiry, then the task could be tagged as Expired.

In this lab, you will:

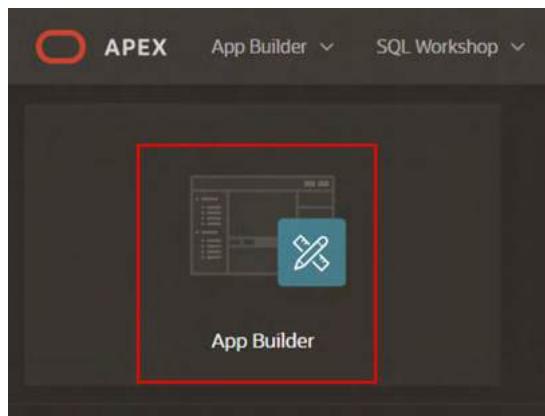
- Obtain a free development environment
- Learn how to create a SQL Script
- Learn how to create a Task definition
- Learn how to create a Page to submit Expense
- Learn how to create a Unified Task List
- Learn how to create Users
- Learn how to create an Email Template
- Learn how to request Information/ Delegate/ Release
- Learn how to set deadlines and expiration

Create the Application

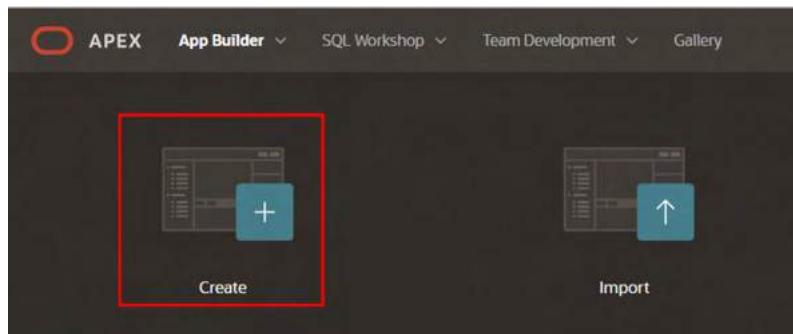
In this lab, you create a new application named Expense Tracker.

1. If you have not already logged in to your Oracle APEX workspace, sign in using the workspace name, email, and password you signed up with.

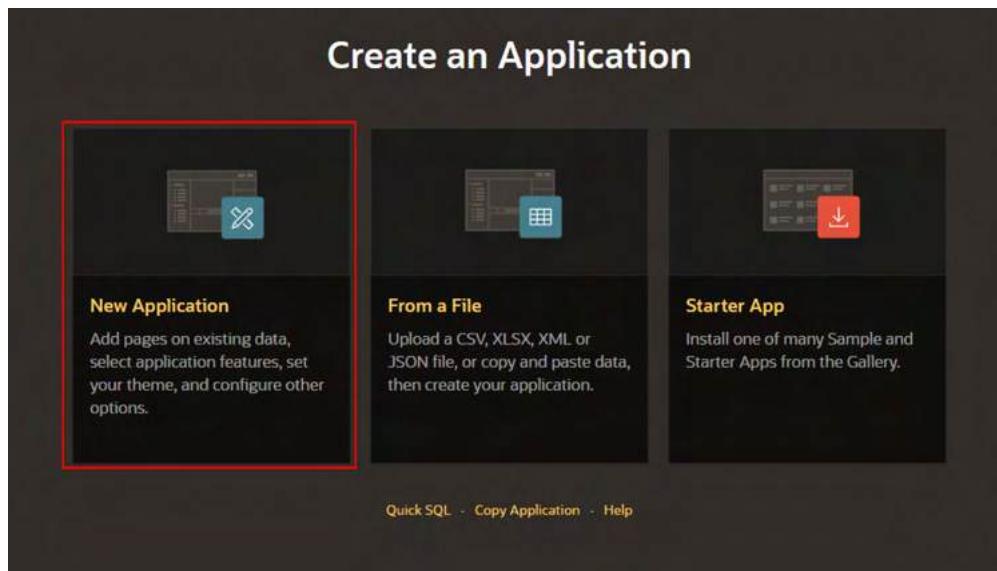
2. At the top left of your workspace, click **App Builder**.



3. On the App Builder page, click **Create**.

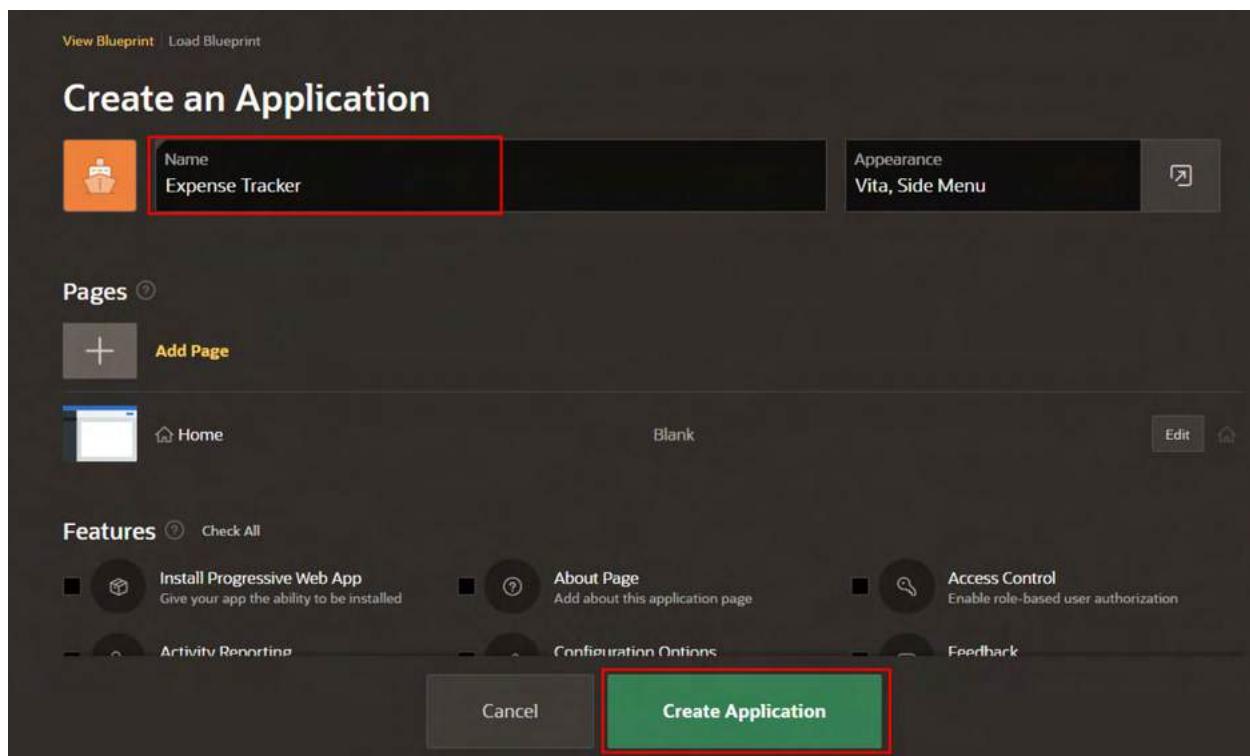


4. Click **New Application**.



5. In the Create an Application Wizard, set Name to **Expense Tracker**.

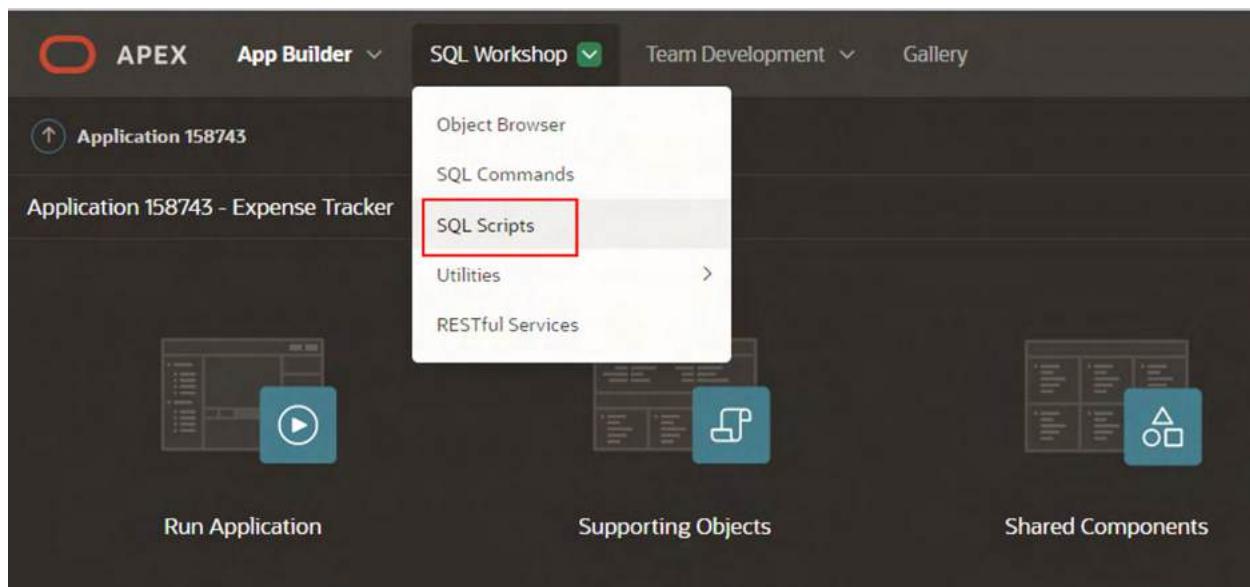
6. Click **Create Application** to create your app and go to the application home page.



Create a SQL Script

In this lab, you create database objects using SQL Script.

7. At the top of the application home page, click **SQL Workshop** and then **SQL Scripts**. The SQL Scripts page appears.



8. Click **Create**. The Script Editor appears.



9. In the Script Name, enter a name for the script - **Employee details and Expense status**.
10. Enter the SQL statements.

Copy the code below and paste it into the code editor:

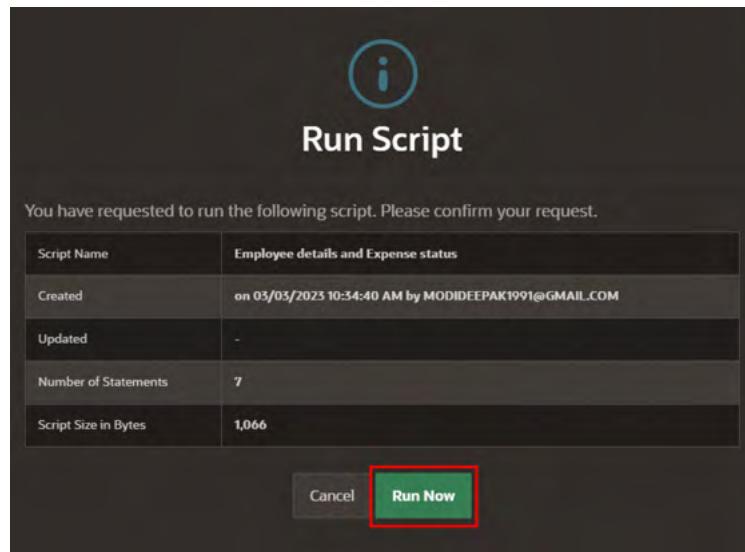
```
CREATE TABLE "EMPLOYEE_DETAILS"
("EMPNO" NUMBER GENERATED BY DEFAULT ON NULL
AS IDENTITY MINVALUE 1 MAXVALUE 99999999999999999999999999999999
INCREMENT BY 1 START WITH 8000 CACHE 20 NOORDER NOCYCLE NOKEEP
NOT NULL ENABLE,
"EMP_NAME" VARCHAR2(100),
"MGR" NUMBER(4,0),
CONSTRAINT "EMP_PK" PRIMARY KEY ("EMPNO"));
insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (10,
'JOHN', 20);
insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (20,
'CLARA',30);
insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (30,
'JANE', 40);
insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (40,
'MATT', null);
commit;
CREATE TABLE "EMP_EXPENSE_REQUEST"
("REQ_ID" NUMBER GENERATED BY DEFAULT ON NULL
AS IDENTITY MINVALUE 1 MAXVALUE 99999999999999999999999999999999
INCREMENT BY 1 START WITH 8000 CACHE 20 NOORDER NOCYCLE NOKEEP
NOT NULL ENABLE,
"EMP_NO" VARCHAR2(10),
"EXPENSE_TYPE" VARCHAR2(100),
"ESTIMATED_COST" NUMBER(8,0),
"UPDATED_BY" VARCHAR2(100),
"STATUS" VARCHAR2(20),
CONSTRAINT "EMP_EXPENSE_REQUEST_PK" PRIMARY KEY ("REQ_ID"));
```

11. Click **Run** and **Run Now**. Now you can see that SQL statements run successfully.

```

1 CREATE TABLE "EMPLOYEE_DETAILS"
2 ("EMPNO" NUMBER GENERATED BY DEFAULT ON NULL
3 AS IDENTITY MINVALUE 1 MAXVALUE 99999999999999999999999999999999
4 INCREMENT BY 1 START WITH 8000 CACHE 20 NOORDER NOCYCLE NOKEEP NOT NULL ENABLE,
5 "EMP_NAME" VARCHAR2(100),
6 "MGR" NUMBER(4,0),
7 CONSTRAINT "EMP_PK" PRIMARY KEY ("EMPNO"));
8 insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (10, 'JOHN', 20);
9 insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (20, 'CLARA', 30);
10 insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (30, 'JANE', 40);
11 insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (40, 'MATT', null);
12 || commit;
13 CREATE TABLE "EMP_EXPENSE_REQUEST"
14 ("REQ_ID" NUMBER GENERATED BY DEFAULT ON NULL
15 AS IDENTITY MINVALUE 1 MAXVALUE 99999999999999999999999999999999
16 INCREMENT BY 1 START WITH 800 CACHE 20 NOORDER NOCYCLE NOKEEP NOT NULL ENABLE,
17 "EMP_NO" VARCHAR2(10),
18 "EXPENSE_TYPE" VARCHAR2(100),
19 "ESTIMATED_COST" NUMBER(8,0),
20 "UPDATED_BY" VARCHAR2(100),
21 "STATUS" VARCHAR2(20),
22 CONSTRAINT "EMP_EXPENSE_REQUEST_PK" PRIMARY KEY ("REQ_ID"));

```



SQL Scripts > Results

Script: Employee details and Expense status Status: Complete

View: Detail Summary Rows: 15 Go

Create App Edit Script

| Number | Elapsed | Statement | Feedback | Rows |
|--------|---------|--|----------------------|------|
| 1 | 0.05 | CREATE TABLE "EMPLOYEE_DETAILS" ["EMPNO" NUMBER GENERATED BY | Table created. | 0 |
| 2 | 0.05 | Insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (1 | 1 row(s) inserted. | 1 |
| 3 | 0.00 | insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (2 | 1 row(s) inserted. | 1 |
| 4 | 0.00 | insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (3 | 1 row(s) inserted. | 1 |
| 5 | 0.01 | insert into EMPLOYEE_DETAILS(empno, emp_name, mgr) values (4 | 1 row(s) inserted. | 1 |
| 6 | 0.00 | commit | Statement processed. | 0 |
| 7 | 0.05 | CREATE TABLE "EMP_EXPENSE_REQUEST" ("REQ_ID" NUMBER GENERATE | Table created. | 0 |

Download

7 Statements Processed

7 Successful

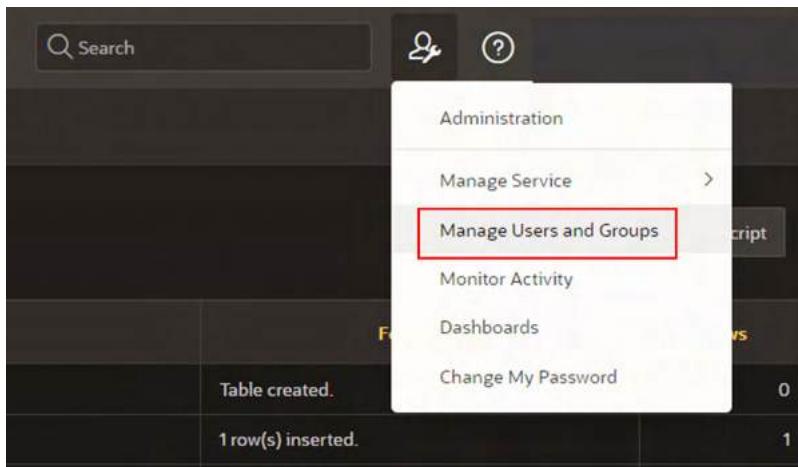
0 With Errors

Row(s) 1 - 7 of 7

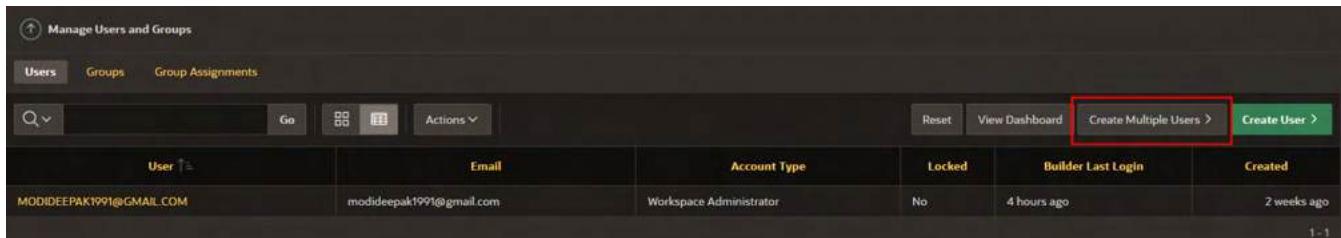
Add Users

In this lab, you create users for multilevel management.

1. Navigate to the Administration icon on the Application home page and select **Manage Users and Groups** from the drop-down list.



2. Click **Create Multiple Users**.



3. Specify the following attributes:

- For a List of Email Addresses: Enter **MATT@oracle.com, JANE@oracle.com, CLARA@oracle.com, JOHN@oracle.com**
- For Usernames: Select **Exclude @ domain as part of the username**.
- For password and Confirm Password: Enter a password of your wish.

4. Click **Next** and **Create Valid Users**.

Create Multiple Users

To create multiple users at once, enter or copy and paste email addresses separated by commas, semicolons, or new lines.
Note that the password you specify will be assigned to each user and users will need to change their passwords upon login.

List of Email Addresses: **MATT@oracle.com, JANE@oracle.com, CLARA@oracle.com, JOHN@oracle.com**

Users: Set username to full email address ?
 Exclude @ domain as part of the username

Account Privileges

Default Schema: **WKSP_APEXHANDSONLAB22** ?

Accessible Schemas (null for all)

Users are workspace administrators: Yes No ?

Users are developers: Yes No ?

Account Privileges

Default Schema: **WKSP_APEXHANDSONLAB22** ?

Accessible Schemas (null for all)

Users are workspace administrators: Yes No ?

Users are developers: Yes No ?

App Builder Access: **No** ?

SQL Workshop Access: **No** ?

Team Development Access: **No** ?

Password

* Password: **.....** Passwords are case sensitive ?

* Confirm Password: **.....** ?

Create Multiple Users

Valid Users

| Username | Email |
|----------|------------------|
| CLARA | CLARA@oracle.com |
| JANE | JANE@oracle.com |
| JOHN | JOHN@oracle.com |
| MATT | MATT@oracle.com |

1 - 4

Invalid Users

No invalid data found.

[«](#)
[Cancel](#)
Create Valid Users

✓ Users Created

You can download Oracle APEX 22.2 here.

 Manage Service
  Manage Users and Groups
  Monitor Activity
  Dashboards
  Utilization Report

Workspace Message

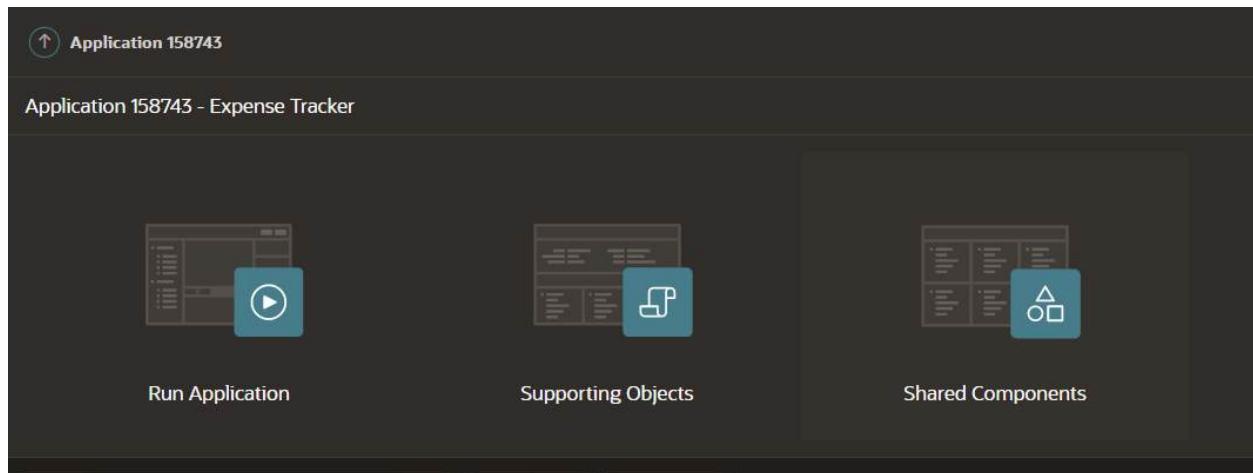
| Service | Users | Activity |
|----------------|----------------------------|-------------------------------|
| Workspace Name | APEX_HANDS_ON_LAB22 | 5 Reporting Timeframe 3 hours |
| Applications | 4 Workspace Administrators | 1 Page Events 269 |
| SQL Scripts | 3 Application Developers | 1 Median Page Time 0.58 |
| Schemas | 1 End Users | 4 Distinct Applications 5 |
| Open Requests | 0 Created Last 24 Hours | 4 Distinct Users 2 |
| | Created Last Week | 4 Distinct Sessions 2 |
| | | Errors 0 |

Create a Task Definition

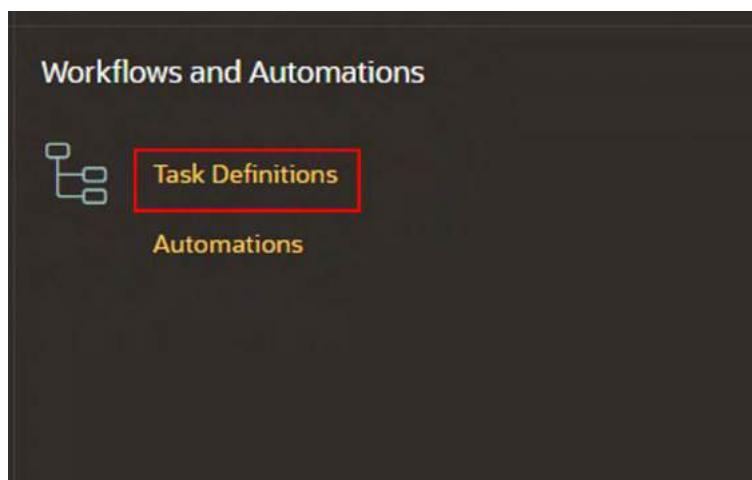
Create a task definition to configure task parameters, participants, actions, and due dates for an expense request.

To create a task definition:

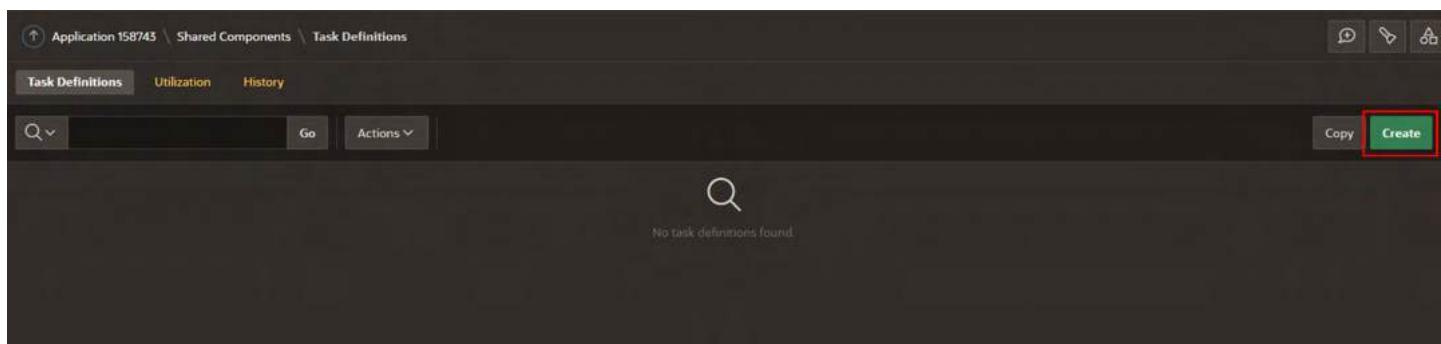
1. In the **App Builder**, navigate to the **Expense Tracker** application and select **Shared Components**.



2. Under Workflows and Automations, select **Task Definitions**.



3. Click **Create**.



4. Specify the task definition name and define the metadata.

- For Name - Enter **Expense Request**.
- For Subject - Enter **&EXPENSE_TYPE. Expense request for &EMPNAME**.
- For Static ID - Enter **EXPENSE_REQUEST**.
- For Priority - Select **2-High**.

Leave Business Administrator and Potential owner blank for now.

Create Task Definition

* Name: Expense Request

* Subject: &EXPENSE_TYPE. Expense request for &EMPNAME

* Static ID: EXPENSE_REQUEST

* Priority: 2-High

Potential Owner:

Business Admin:

Cancel Create

5. Click **Create**.

6. Under the **Settings** Section:

- For Task details Page Number - Click the **Create Task Details Page** button, then click **OK**.
- For Actions Source - Select SQL Query.
- For Actions SQL query - Copy the code below and paste it into the code editor:

```
select EMP_NAME from employee_details where EMPNO =(select MGR  
from employee_details where EMPNO=(select EMPNO from  
employee_details where EMP_NAME=:APP_USER))
```

Task Definition: Expense Request

Settings

Subject: &EXPENSE_TYPE. Expense request for &EMPNAME.

Priority: 2-High

Task Details Page URL: f?p=&APP_ID.2:&SESSION.=&DEBUG.RP:2:P2_TASK_ID:&TASK_ID.

Actions Source: SQL Query (highlighted)

Actions SQL Query:

```
1  oyee_details where EMPNO =(select MGR from employee_details where EMPNO=(select EMPNO from employee_details where EMP_NAME=:APP_USER))
```

7. Under Participants Section - Select Participants to assign additional people to the Task Definition.

- Click Add Row.
- For Participant Type - Select **Potential Owner**
- For Value Type - Select **SQL Query**
- For Value - Copy the code below and paste it into the code editor:

```
select EMP_NAME from employee_details where EMPNO =(select MGR
from employee_details where EMPNO=(select EMPNO from
employee_details where EMP_NAME=:APP_USER))
```

Task Definition: Expense Request

Participants

| Participant Type | Value Type | Value |
|------------------|------------|--|
| Potential Owner | SQL Query | select EMP_NAME from employee_details where EMPNO =(select MGR fro... Total 1 |

8. Click the task definition - **Expense Request** to continue editing.

The screenshot shows a table with columns: Name, Subject, Priority, Participants, Parameters, Actions, Last Updated By, and Last Updated On. The 'Expense Request' row is selected, highlighted with a red border. The 'Actions' column contains a 'Copy' button and a 'Create' button. A message at the top left says 'Changes applied. Participants saved.'

| Name | Subject | Priority | Participants | Parameters | Actions | Last Updated By | Last Updated On |
|-----------------|---|----------|--------------|------------|---------|--------------------------|-----------------|
| Expense Request | &EXPENSE_TYPE: Expense request for &EMPNAME | 2 | 1 | 0 | 0 | MODIDEEPAK1991@GMAIL.COM | 03/03/2023 |

9. Under **Parameters** Section - Select Add Row and fill in the four parameter fields listed below:

Table 1: Extend Application Capabilities | Lab 1: Manage Approvals Component

| Static ID | Label | Data Type |
|----------------|----------------|-----------|
| ESTIMATED_COST | Estimated Cost | String |
| EXPENSE_STATUS | Expense Status | String |
| EXPENSE_TYPE | Expense Type | String |
| REQ_ID | Req Id | String |
| Static ID | Label | Data Type |

The screenshot shows the 'Parameters' tab selected in the 'Task Definition: Expense Request' interface. It displays a table with columns: Static ID, Label, Data Type, Required, Visible, and Comment. Four rows are present: ESTIMATED_COST (Label: Estimated Cost, Data Type: String, Required: Yes, Visible: Yes), EXPENSE_STATUS (Label: Expense Status, Data Type: String, Required: Yes, Visible: Yes), EXPENSE_TYPE (Label: Expense Type, Data Type: String, Required: Yes, Visible: Yes), and REQ_ID (Label: Req Id, Data Type: String, Required: Yes, Visible: Yes). The 'Add Row' button is visible at the top of the table.

| Static ID | Label | Data Type | Required | Visible | Comment |
|----------------|----------------|-----------|----------|---------|---------|
| ESTIMATED_COST | Estimated Cost | String | Yes | Yes | |
| EXPENSE_STATUS | Expense Status | String | Yes | Yes | |
| EXPENSE_TYPE | Expense Type | String | Yes | Yes | |
| REQ_ID | Req Id | String | Yes | Yes | |

10. Under the **Actions** section, click the **Add Action** button.

The screenshot shows the Oracle Application Builder interface for creating a task definition. The title bar indicates the path: Application 158743 \ Shared Components \ Task Definitions \ Expense Request. The main area is titled "Task Definition: Expense Request". Below it, there's a toolbar with buttons for Show All, Name, Settings, Deadline, Participants, Parameters, Actions, Comments, and Last Updated. The "Actions" button is highlighted with a red box. To the right of the toolbar are buttons for Cancel, Delete, and Apply Changes. The main content area is titled "Actions" and contains a table with columns: Name, Outcome, Execution Sequence ↑↓, and Action Type. A search bar and a "Go" button are at the top of the table. A red box highlights the "Add Action" button in the top right corner of the table area. A message below the table says "Click Add Action to create the first Action."

Specify the following:

- For Name - Enter **CREATE_EXPENSE_REPORT_ENTRY**.
- For Type - Select Execute Code.
- On Event - Select Create.
- For Code: Copy the code below and paste it into the code editor:

```
declare
    l_req_id number;
begin
    if :APP_USER = :EMP_NAME then --this is the original initiator
        l_req_id := :APEX$TASK_ID;
    -- create a new record in the Employee Expense Request table
    EMP_EXPENSE_REQUEST
    insert into EMP_EXPENSE_REQUEST values
    (to_number(l_req_id),
    :EMPNO,
    :EXPENSE_TYPE,
    :ESTIMATED_COST,
    '',
    'PENDING');
    end if;
end;
```

- Click **Create** to save the Create Event action.

Action

- Name: CREATE_EXPENSE_REPORT_ENTRY
- Type: Execute Code
- Execution Sequence: 10
- On Event: Create

Code

Location: Local Database REST Enabled SQL Service

Language: PL/SQL

```

1 declare
2   l_req_id number;
3 begin
4   if :APP_USER = :EMP_NAME then --this is the original initiator
5     l_req_id := :APEX$TASK_ID;
6   -- create a new record in the Employee Expense Request table EMP_EXPENSE_REQUEST
7   insert into EMP_EXPENSE_REQUEST values
8   (to_number(l_req_id),
9    :EMPNO,
10   :EXPENSE_TYPE,

```

Code

Location: Local Database REST Enabled SQL Service

Language: PL/SQL

```

1 declare
2   l_req_id number;
3 begin
4   if :APP_USER = :EMP_NAME then --this is the original initiator
5     l_req_id := :APEX$TASK_ID;
6   -- create a new record in the Employee Expense Request table EMP_EXPENSE_REQUEST
7   insert into EMP_EXPENSE_REQUEST values
8   (to_number(l_req_id),
9    :EMPNO,
10   :EXPENSE_TYPE,

```

Examples

Error Handling

11. To Add the next action, Click the **Add Action** button.

Specify the following:

- For Name - Enter **NEXT_APPROVER_OR_UPDATE_STATUS**.
- For Type - Select Execute Code.
- On Event - Select Complete.
- For Outcome : Select Approved.
- For Code: Copy the code below and paste it into the code editor:

```

declare
  l_mgr number;
  l_task_id number;
  l_request_id number;
  l_req_status varchar2(10) := 'PENDING';
begin
  if :APP_USER = :MGR_NAME then --this is the first approver

```

```

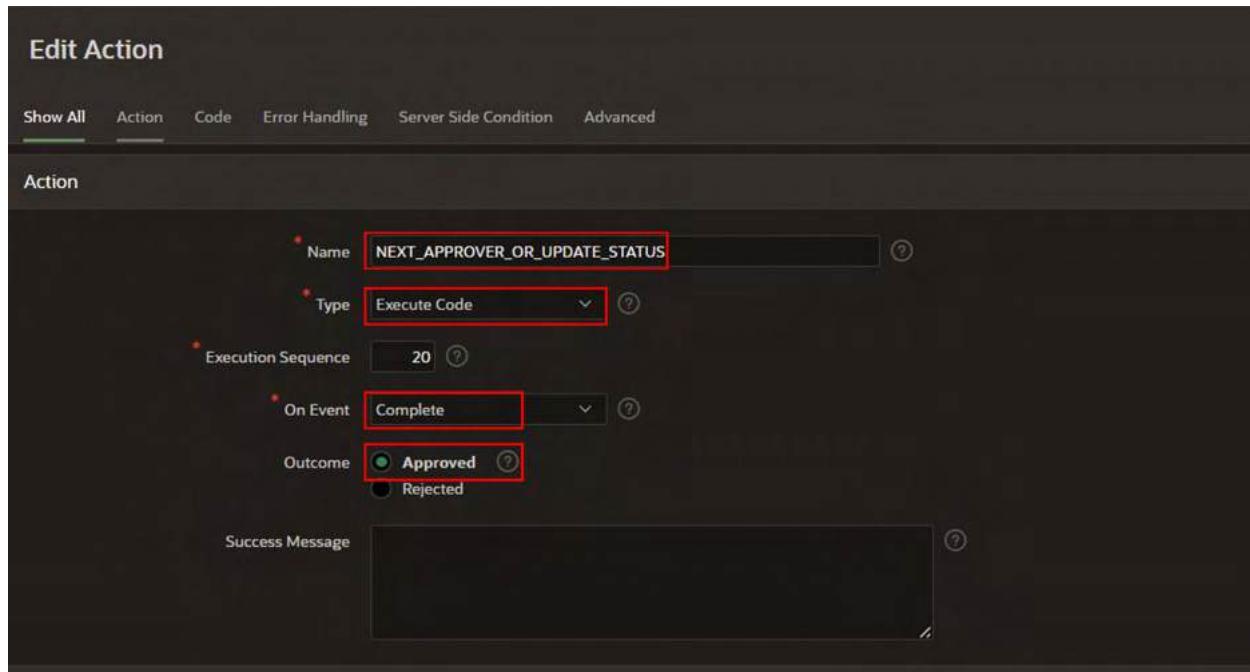
-- set the request id to be the id of the task created when the
request was submitted
l_request_id := :APEX$TASK_ID;
else
-- this is an intermediate approver. Set the request-id from
the corresponding task parameter value
l_request_id := :REQ_ID;
end if;
if :ESTIMATED_COST < 50000 then -- the approval is complete

update EMP_EXPENSE_REQUEST set status = 'APPROVED',
updated_by=updated_by||'->'||:APP_USER
where req_id = l_request_id and emp_no=:APEX$TASK_PK;

l_req_status := 'APPROVED';
else -- the request needs to go through another level of
Approval
-- updated the request record with details of the current
approver in the chain of approvers
update EMP_EXPENSE_REQUEST set updated_by = updated_by||'-
>'||:APEX$TASK_OWNER
where req_id = l_request_id
and emp_no=:APEX$TASK_PK;
-- create a new task assigned to the manager of the current
approver
l_task_id := apex_approval.create_task(
    p_application_id => :APP_ID,
    p_task_def_static_id => 'EXPENSE_REQUEST',
    p_initiator => :EMP_NAME, -- ensure initiator is the
original requestor and not the current task owner
    p_parameters => apex_approval.t_task_parameters(
        1 => apex_approval.t_task_parameter(static_id =>
'EXPENSE_TYPE', string_value => :EXPENSE_TYPE),
        2 => apex_approval.t_task_parameter(static_id =>
'ESTIMATED_COST', string_value => :ESTIMATED_COST),
        3 => apex_approval.t_task_parameter(static_id =>
'REQ_ID', string_value => l_request_id),
        4 => apex_approval.t_task_parameter(static_id =>
'STATUS', string_value => l_req_status)
    ),
    p_detail_pk => :APEX$TASK_PK
);
end if;
end;

```

- Click **Create** to add action.



12. Again, click the **Add Actions** button.

Specify the following:

- For Name - Enter UPDATE_REQUEST_STATUS.
- For Type - Select Execute Code.
- On Event - Select Complete.
- For Outcome - Select Rejected.
- For Code: Copy the code below and paste it into the code editor:

```
declare
    l_mgr number;
    l_task_id number;
    l_request_id number;
    l_req_status varchar2(10) := 'PENDING';
begin
    select mgr into l_mgr from employee_details where
emp_name=:APP_USER;
    if :APP_USER = :MGR_NAME then --this is the first approver
        l_request_id := :APEX$TASK_ID;
    else
        l_request_id := :REQ_ID;
    end if;-- the request is complete and rejected.
```

```

update EMP_EXPENSE_REQUEST set status = 'REJECTED',
updated_by=updated_by||'->'||:APP_USER
where req_id = l_request_id and emp_no=:APEX$TASK_PK;
end;

```

- Click **Create** and **Apply Changes**.

The screenshot shows the 'Edit Action' page for the 'Expense Request' task definition. The action is named 'UPDATE_REQUEST_STATUS', has a type of 'Execute Code', an execution sequence of 30, and triggers on 'Complete'. The outcome is set to 'Rejected'. A success message field is present but empty.

The screenshot shows the 'Task Definition: Expense Request' page. It lists three actions: 'NEXT_APPROVER_OR_UPDATE_STATUS' (Approved, Sequence 20), 'UPDATE_REQUEST_STATUS' (Rejected, Sequence 30), and 'CREATE_EXPENSE_REPORT_ENTRY' (Sequence 10). The 'UPDATE_REQUEST_STATUS' action is highlighted with a red border.

| Action | Name | Outcome | Execution Sequence | Action Type |
|--------------------|--------------------------------|----------|--------------------|--------------|
| On Event: Complete | NEXT_APPROVER_OR_UPDATE_STATUS | Approved | 20 | Execute Code |
| On Event: Complete | UPDATE_REQUEST_STATUS | Rejected | 30 | Execute Code |
| On Event: Create | CREATE_EXPENSE_REPORT_ENTRY | | 10 | Execute Code |

Create a Page to Apply for Expense

Add a page to submit an Expense request.

1. Click **Application ID**, Application home page appears.

The screenshot shows the Oracle Application home page. At the top, there is a breadcrumb navigation: Application 158743 \ Shared Components \ Task Definitions \ Expense Request. Below the breadcrumb, the title "Task Definition: Expense Request" is displayed. The background is dark, and the text is white or light-colored.

2. Click **Create Page**. The Create a Page Wizard appears.

The screenshot shows the Oracle Application home page again. This time, the "Create Page" button in the bottom right corner of the main content area is highlighted with a red box. The rest of the interface, including the menu bar and toolbars, remains the same.

3. Select **Blank page** under component and click **Next**

The screenshot shows the "Create a Page" wizard. The title bar says "Create a Page". Below it is a tab bar with "Component", "Feature", and "Legacy Pages". The "Component" tab is selected. The main area displays various page components as icons: Blank Page (highlighted with a green border), Calendar, Cards, Chart, Dashboard, Faceted Search, Smart Filters, Form, Interactive Grid, Interactive Report, Map, Search Page, Master Detail, Classic Report, and Plug-in Page. At the bottom left are "Cancel" and "Help" buttons, and at the bottom right is a "Next >" button.

4. Specify Blank Page Attributes.

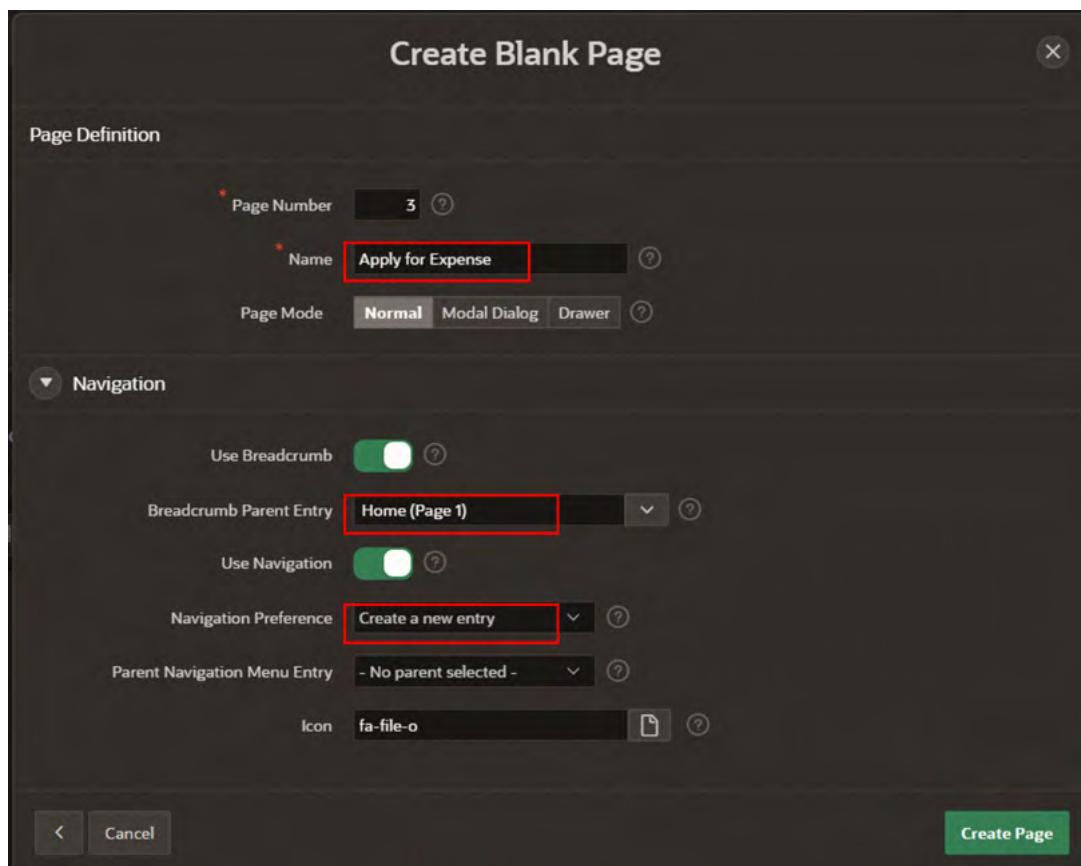
Under the **Page Definition** section:

- For Page Number - Type 3
- For Name - Enter **Apply for Expense**.
- For Page Mode - Select Normal.

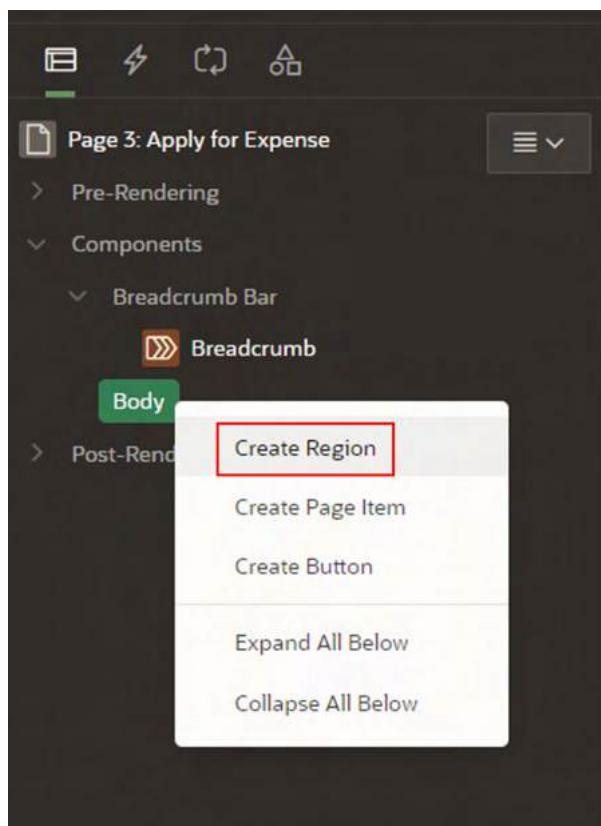
Under the **Navigation** section:

- For Use Breadcrumb - Select On.
- For Breadcrumb Parent Entry - Select Home(Page 1).
- For Use Navigation - Select On.
- For Navigation Preference - Select **Create a new Entry**.
- For Icon - Enter `fa-file-o`.

Click Create Page.



5. In the left pane, select the Rendering tab. Right-click **Body**, select **Create Region**.



6. In the Property Editor, edit the appropriate attributes:

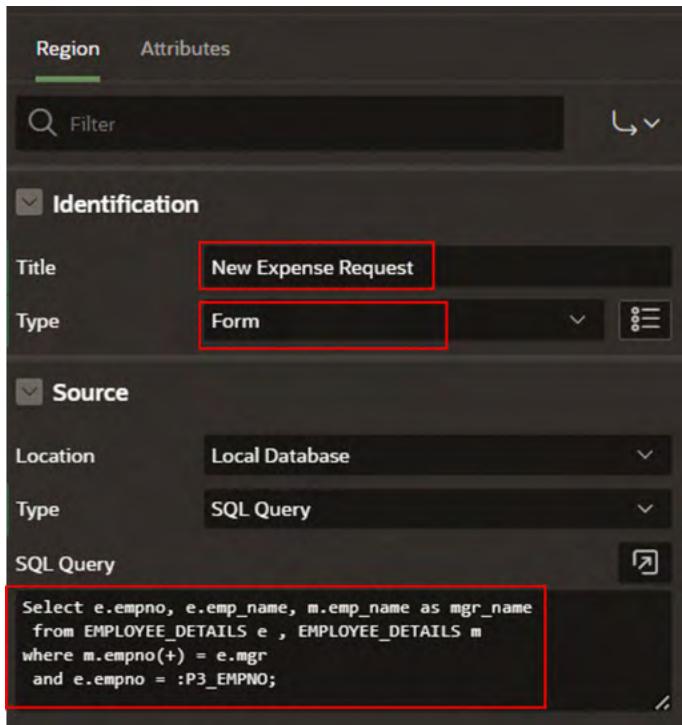
Under **Identification** section:

- For Title: Enter **New Expense Request**.
- For Type: Select Form.

Under **Source** section:

- For Type: Select **SQL Query**.
- For SQL query: Copy the code below and paste it into the code editor:

```
Select e.empno, e.emp_name, m.emp_name as mgr_name  
  from EMPLOYEE_DETAILS e , EMPLOYEE_DETAILS m  
 where m.empno(+) = e.mgr  
   and e.empno = :P3_EMPNO;
```



7. Now, right-click the region (**New Expense Request**) and select Create Page Item.

- For Name - Enter **P3_EXPENSE_TYPE**.
- For Type – Select, **Select List**.

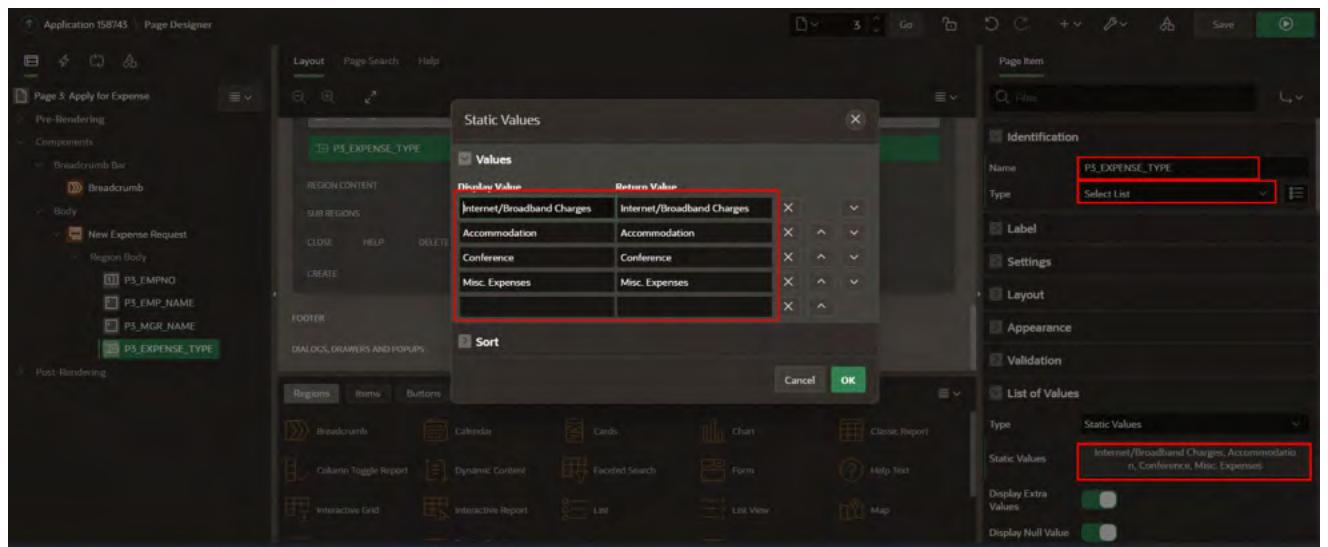
Under **List of Values** section:

- For Type - Select **Static values**.
- For Static Values - Enter below list and click **OK**.

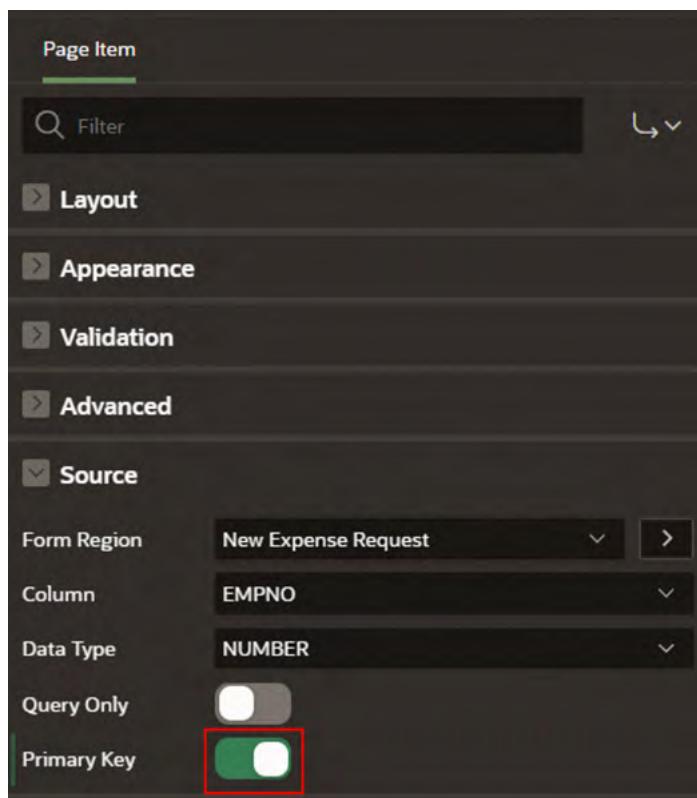
Table 2: Extend Application Capabilities | Lab 1: Manage Approvals Component

| Display Value | Return Value |
|----------------------------|----------------------------|
| Internet/Broadband Charges | Internet/Broadband Charges |
| Accommodation | Accommodation |
| Conference | Conference |
| Misc. Expenses | Misc. Expenses |
| Display Value | Return Value |

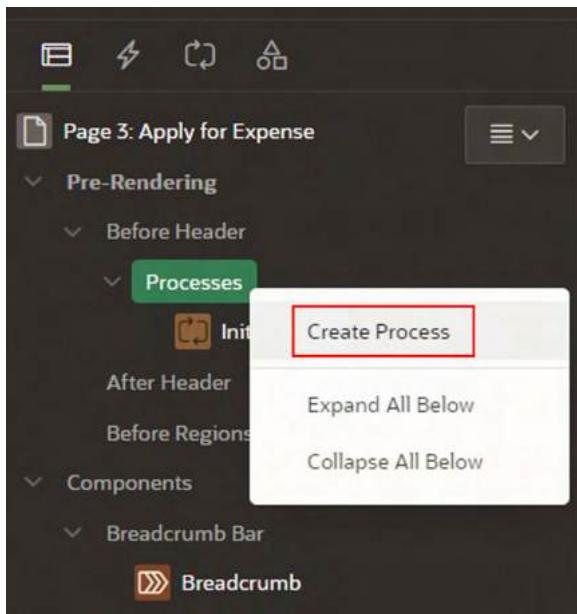
- For Null Display Value - Enter --Select Expense Type--



- Right-click region (**New Expense Request**) and select Create Page Item.
 - For Name - Enter **P3_ESTIMATED_COST**.
 - For Type - Select Number Field.
- Select the **P3_EMPNO** page item and enable the primary key under the **Source** section and Click **Save** to apply changes.



10. On the Rendering tab (left pane), under Pre-Rendering, right-click **Before Header** and click **Create Process**.



In the Property Editor, enter the following:

- For Name - Type **Fetch Employee Details for User**.
- For Type - Select Execute code.
- For PL/SQL Code - Enter the following PL/SQL code:

```
select empno into :P3_EMPNO from employee_details where  
emp_name=:APP_USER;
```

Process

Filter

Identification

Name: Fetch Employee Details for User

Type: Execute Code

Editable Region: - Select -

Source

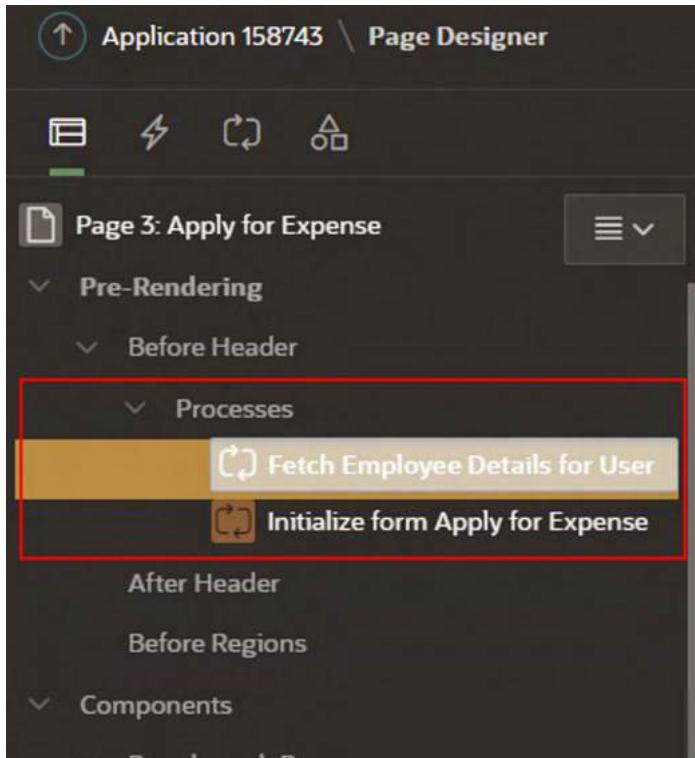
Location: Local Database

Language: PL/SQL

PL/SQL Code:

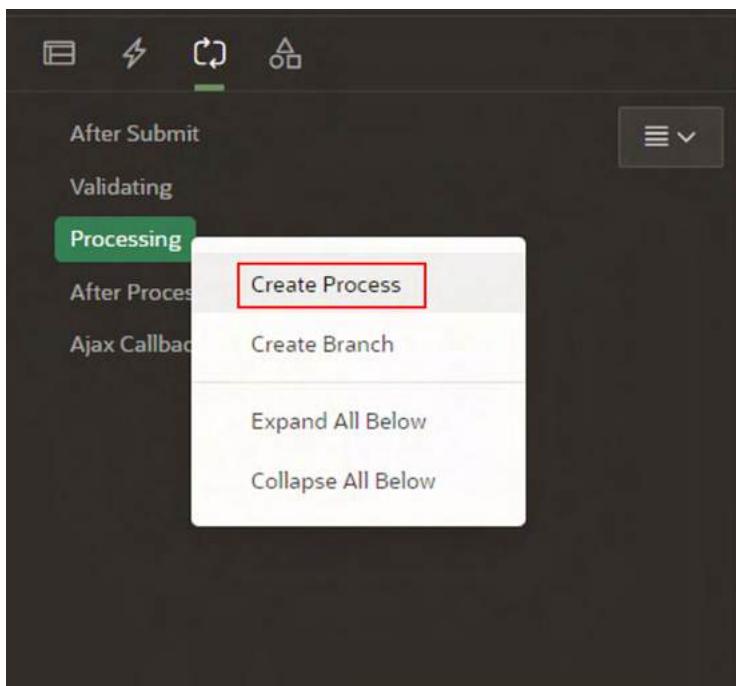
```
select empno into :P3_EMPNO from employee_details where  
emp_name=:APP_USER;
```

- For Sequence - Enter 5.



11. Click **Save**.

12. Now add a process on the **Processing** tab to submit a request. Right-click Processing and select **Create Process**.

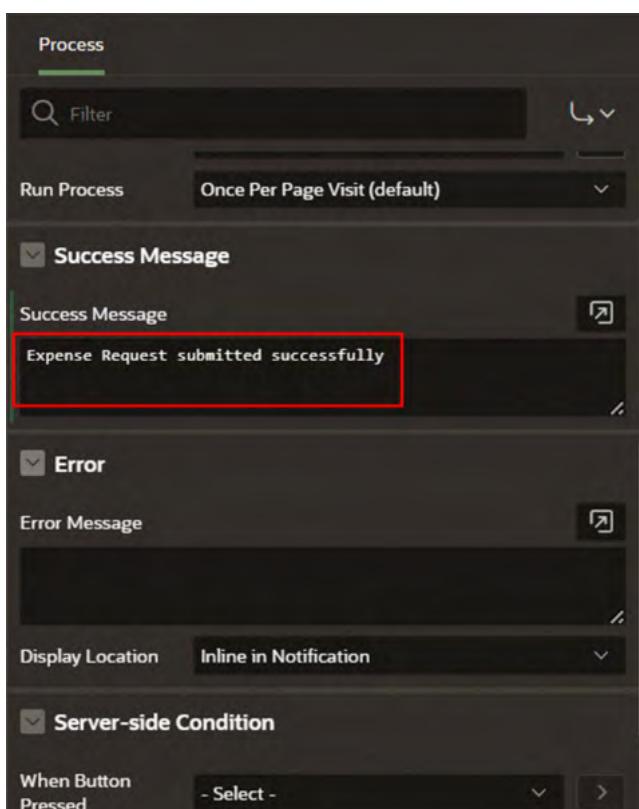
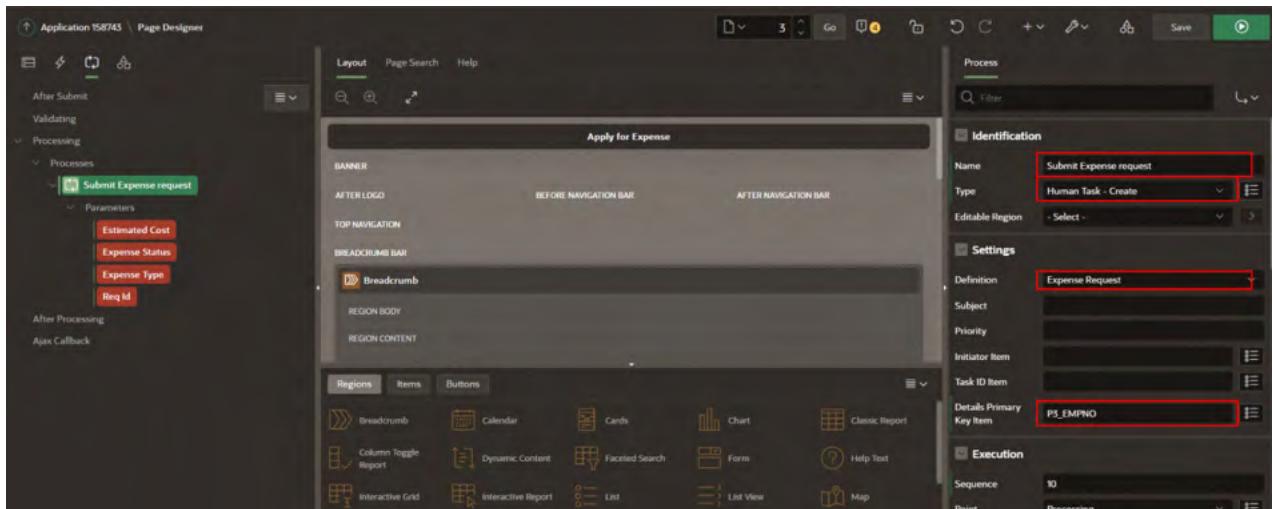


In the Property Editor, enter the following:

- For Name - Type **Submit Expense request**.
- For Type - Select **Human Task – Create**.

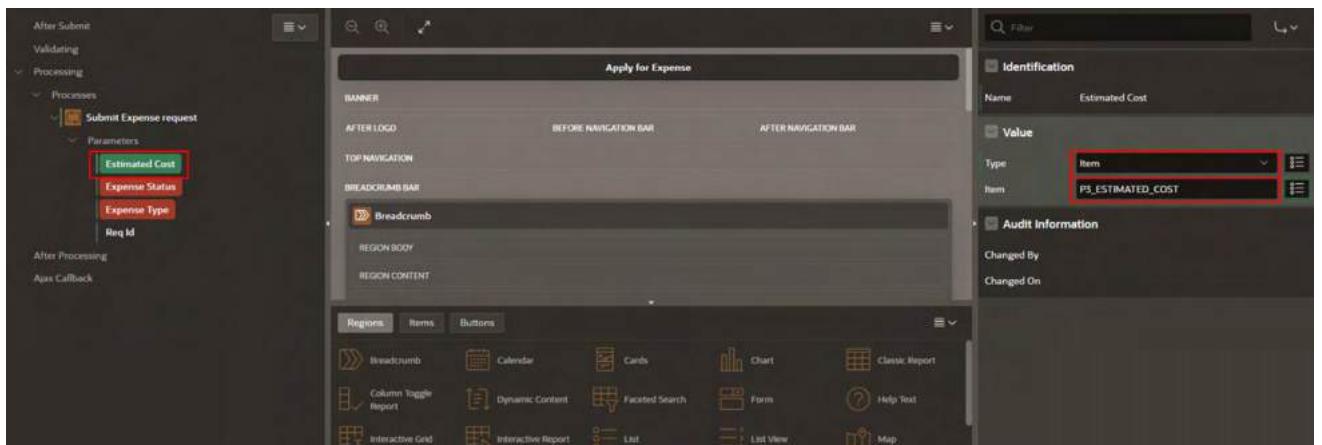
Under the **Settings** section:

- For Definition - Select **Expense request**.
- For Details Primary key Element: Select **P3_EMPNO**
- For Success Message: Type **Expense Request submitted successfully**.



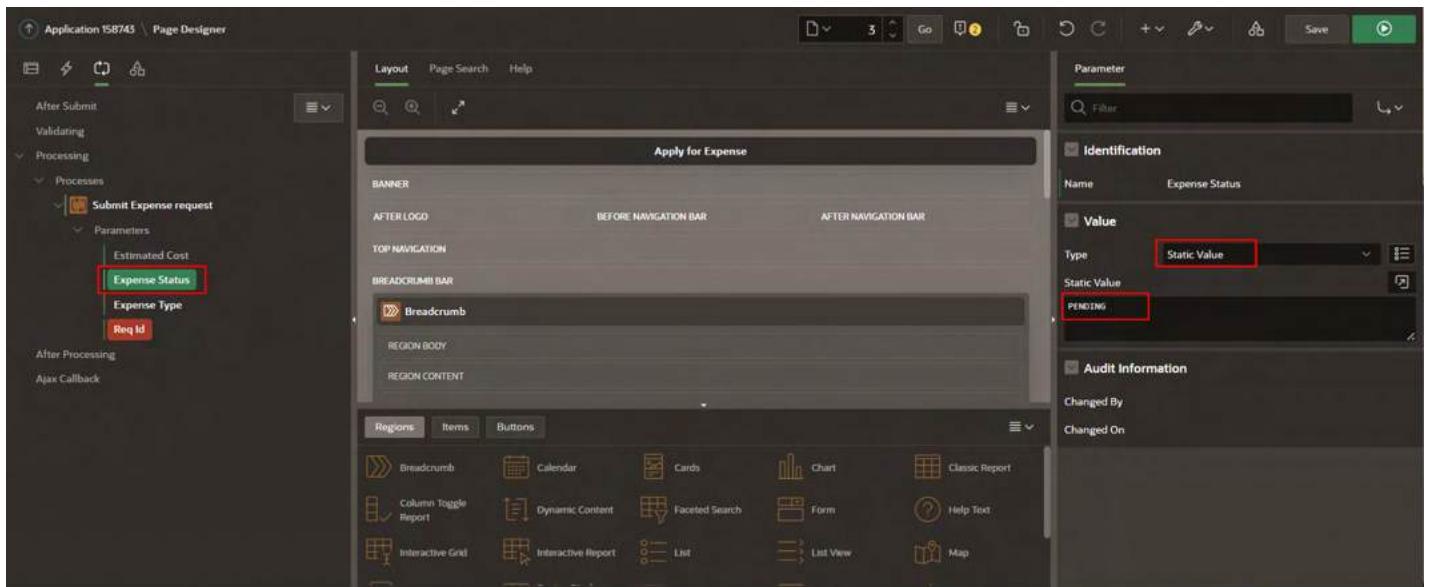
13. Under Parameters for the Submit Expense request process:

- For Estimated Cost, enter the following:
 - For Type - Select ITEM.
 - For Value - Select **P3_ESTIMATED_COST**.



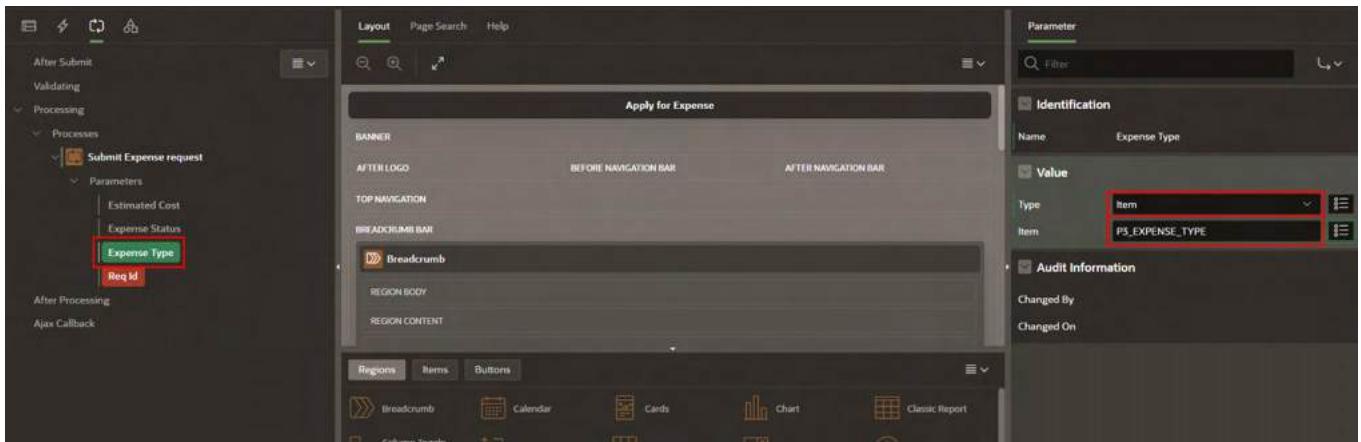
- For Expense Status, enter the following:

- For Type - Select Static Value.
- For Value - Type **PENDING**.



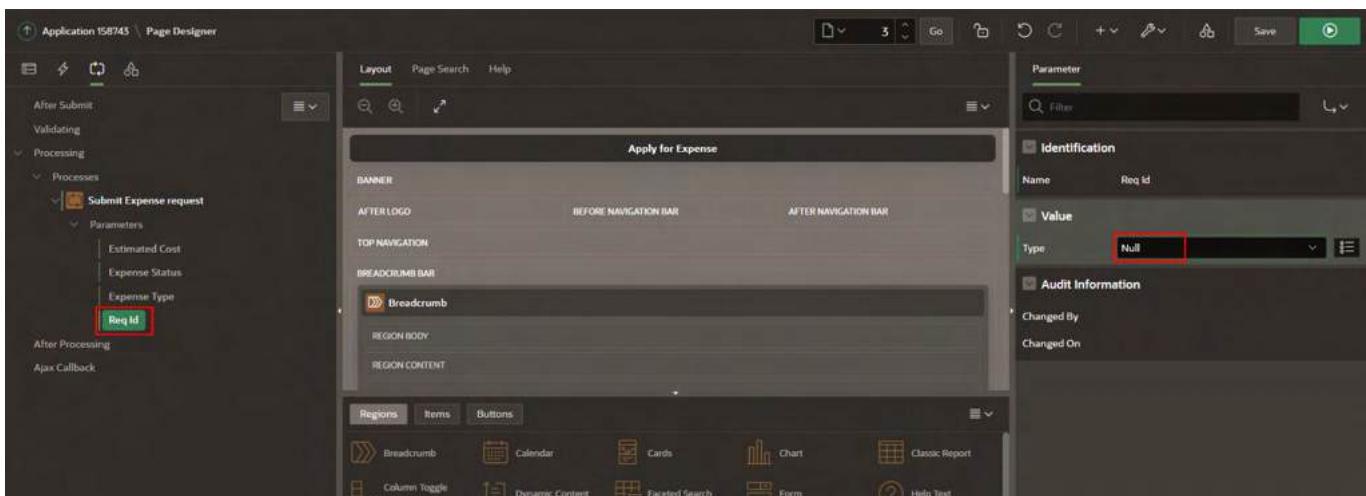
c. For Expense Type, enter the following:

- For Type - Select ITEM.
- For Value - Select **P3_EXPENSE_TYPE**.



d. For Request ID, enter the following:

- For Type - Select NULL.

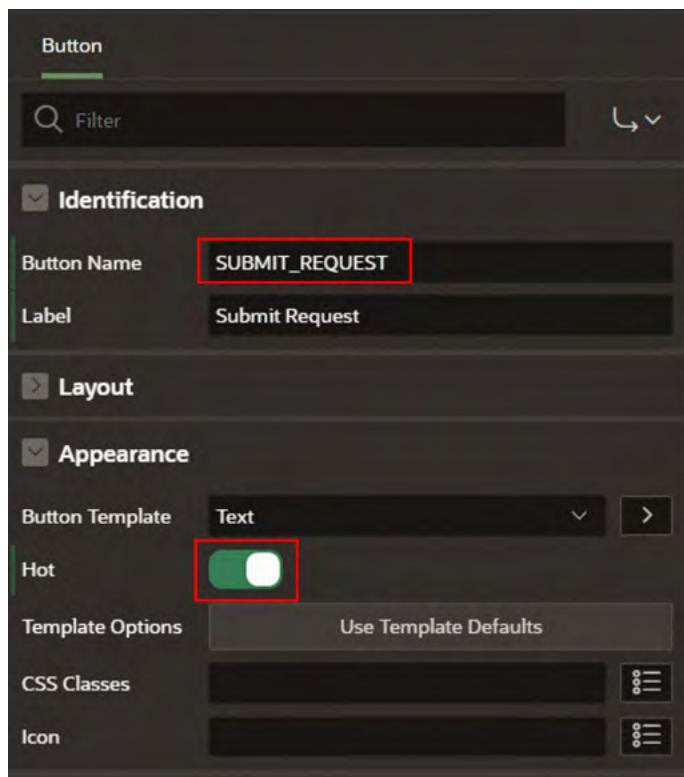


14. Click **Save**

15. On the **Rendering** tab, right-click Body and select **Create Button**.

In the Property Editor, enter the following:

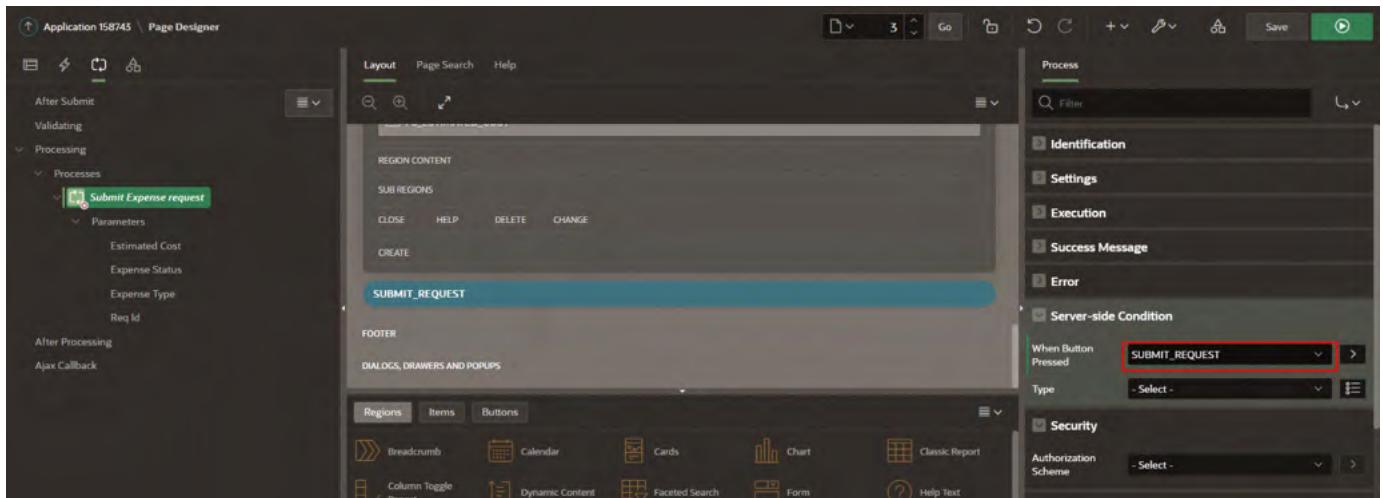
- For Button Name - Type **SUBMIT_REQUEST**.
- For Hot - Select On.



16. Navigate to the **Processing** tab, select the **Submit Expense Request** process.

Under the **Server-side Condition** section:

- For When Button Pressed : Select **SUBMIT_REQUEST**.

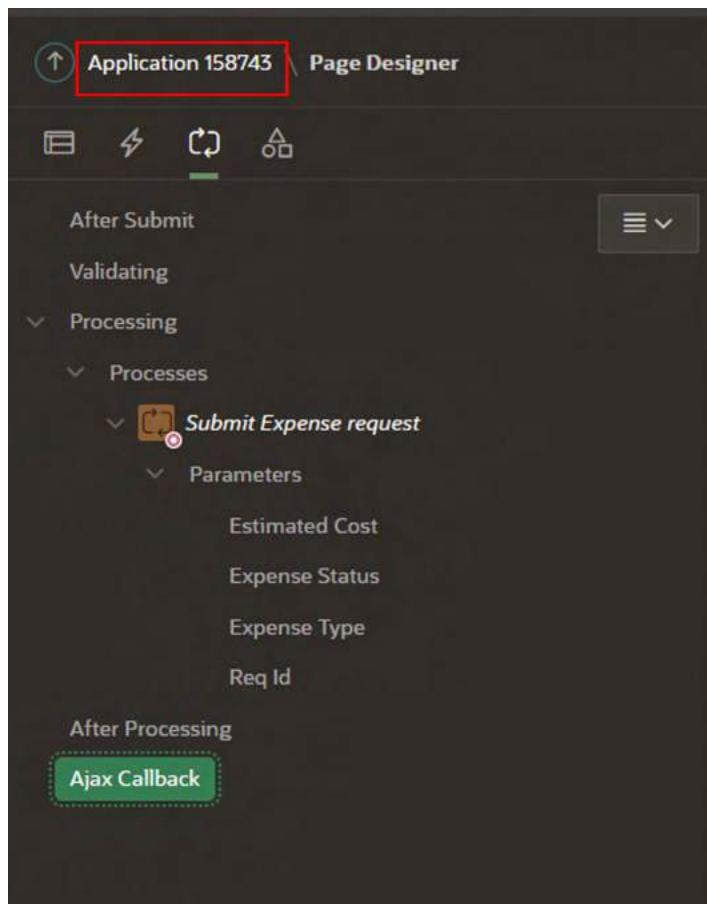


17. Click **Save**.

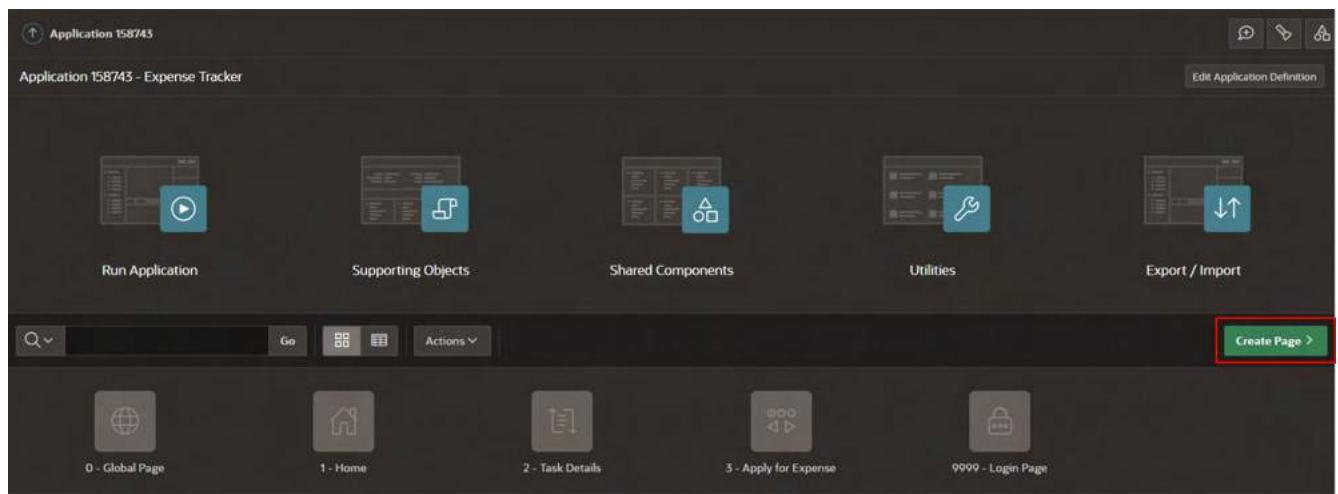
Create Unified Task Lists

Add a Unified Task list page to see the submitted expense request list by a requestor and the Approval list approved or rejected by the approver.

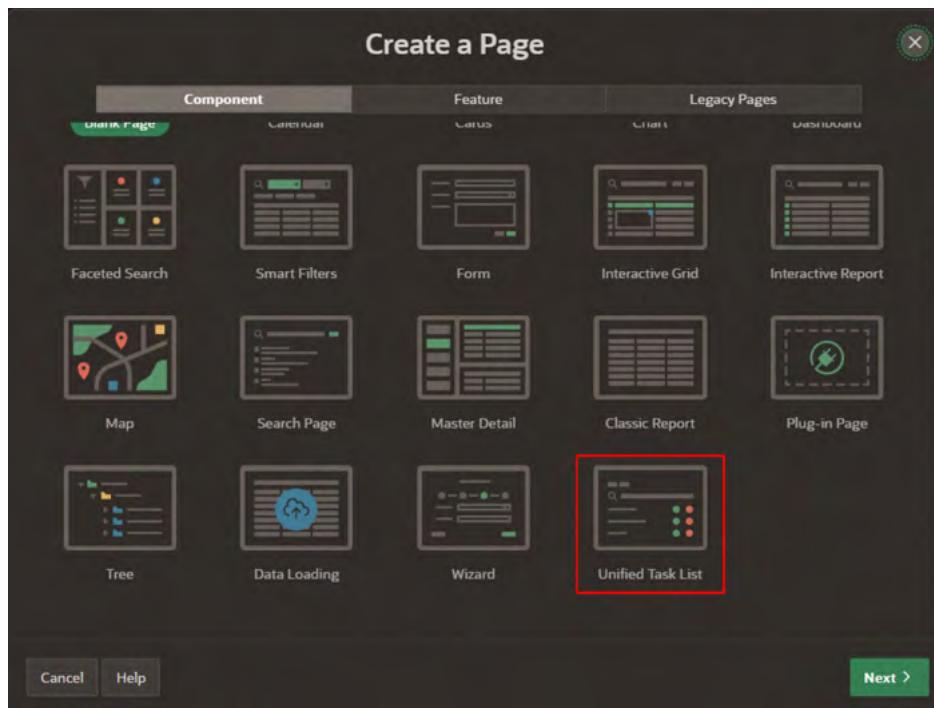
1. Click Application ID on the right-above corner of the page designer. The Application Home page appears.



2. Click Create Page.



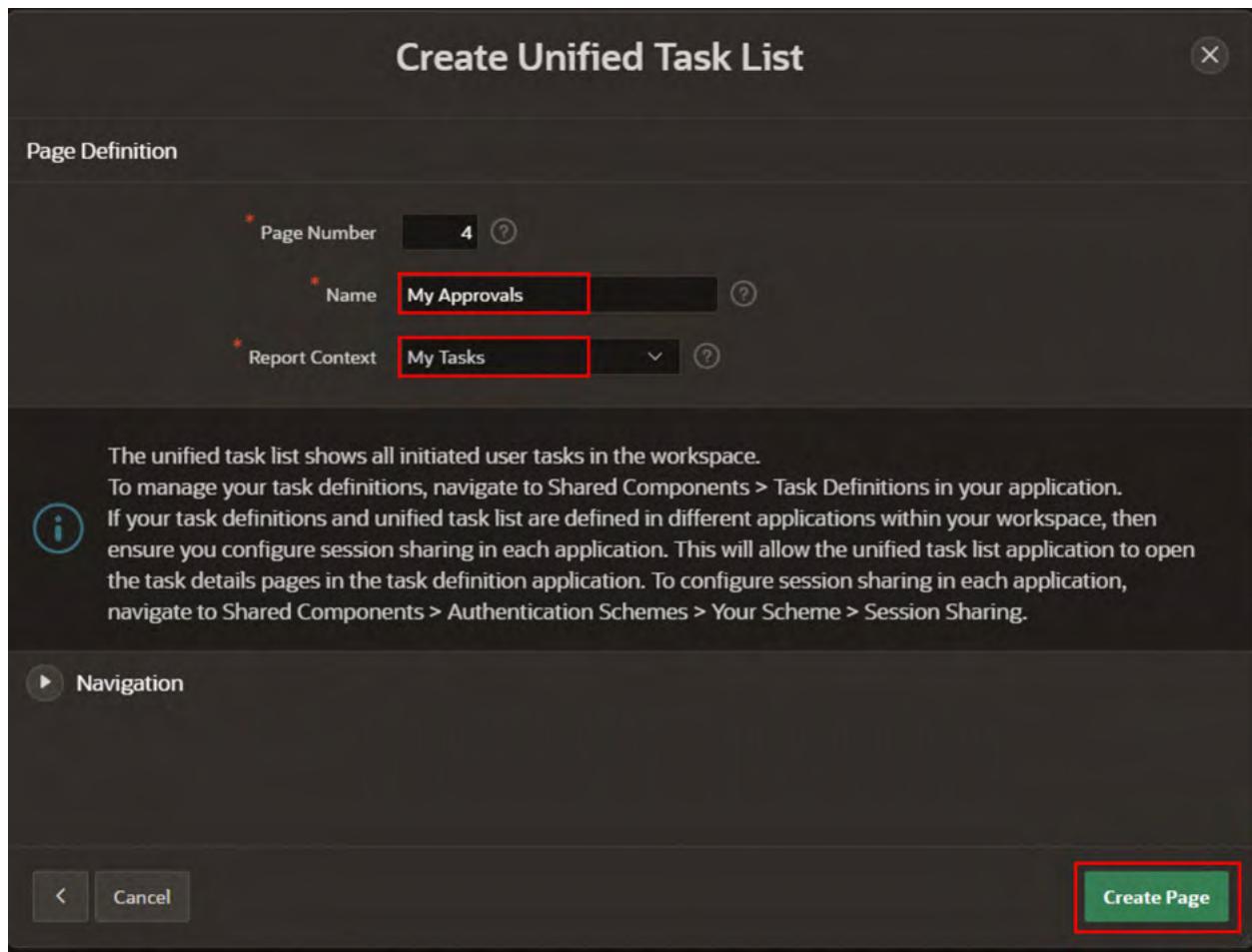
3. Under Components, select Unified Task List and click **Next**.



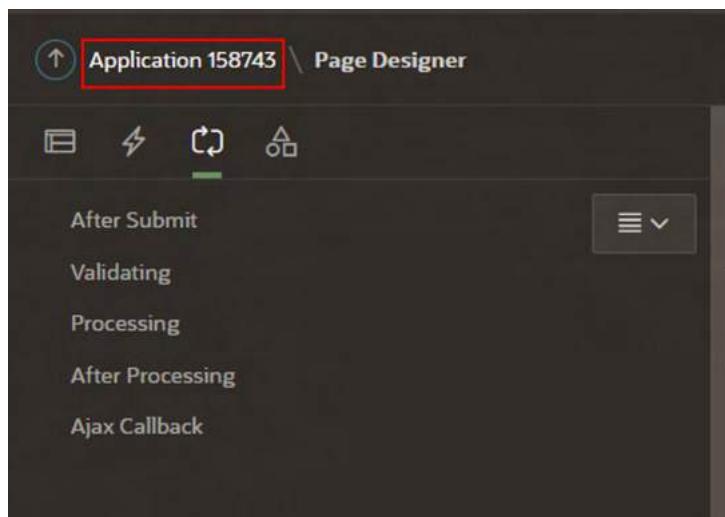
4. Specify the following page attributes:

- For Page Number - Type **4**.
- For Name - Type **My Approvals**.
- For Report Context - Select **My Tasks**.

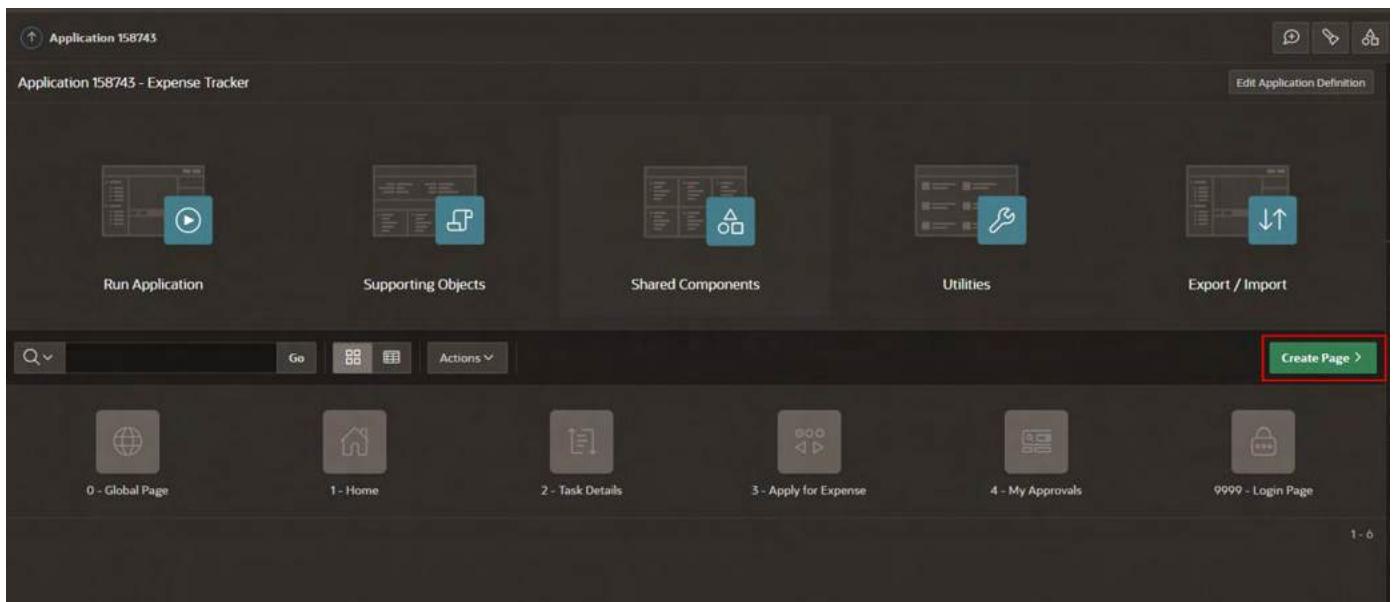
Click **Create Page**. A unified Task List page is created.



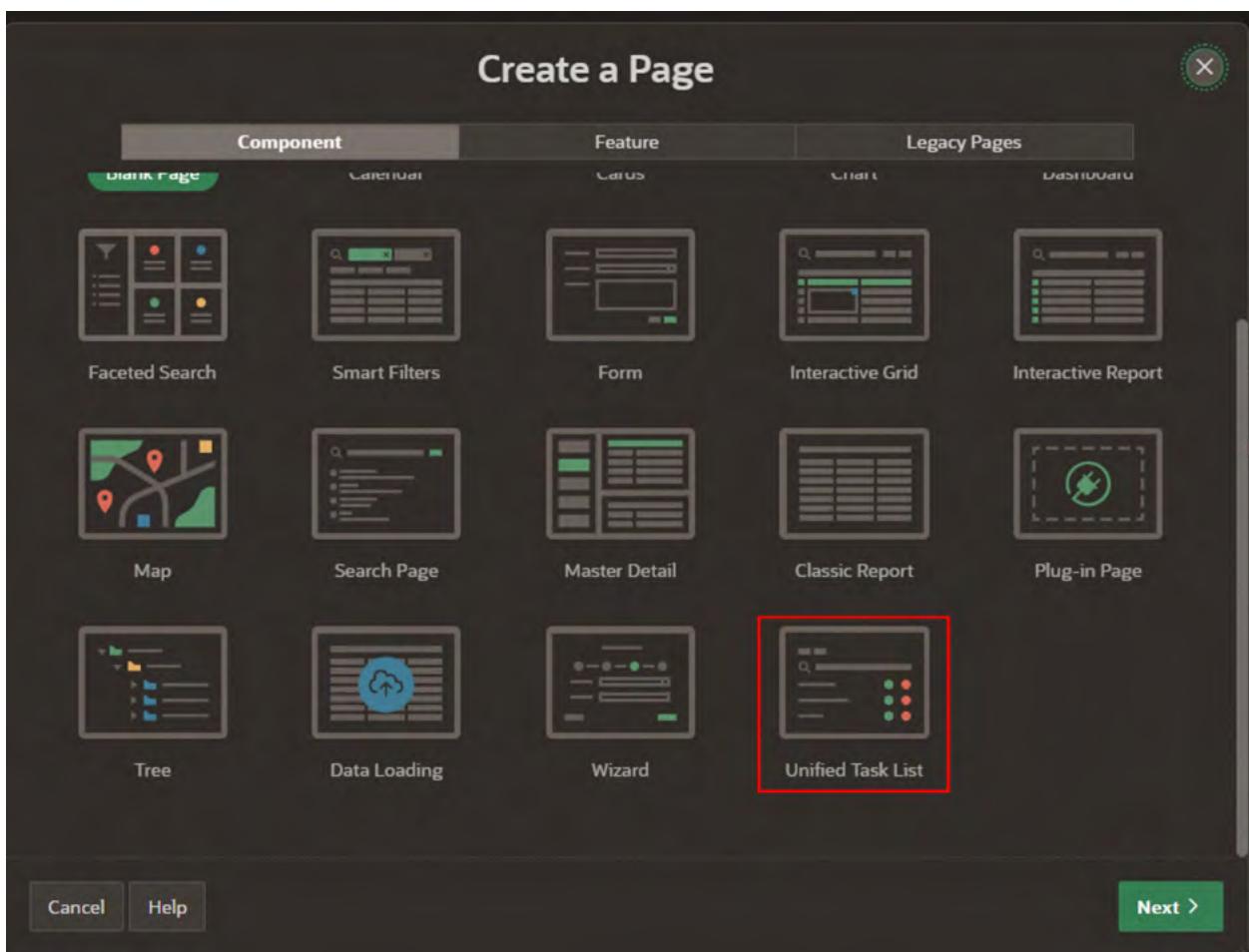
5. To create another Unified Task list page, again, click Application ID on the right-above corner of the page designer. The Application Home page appears.



6. Click the **Create Page** button. The Create a Page Wizard appears.



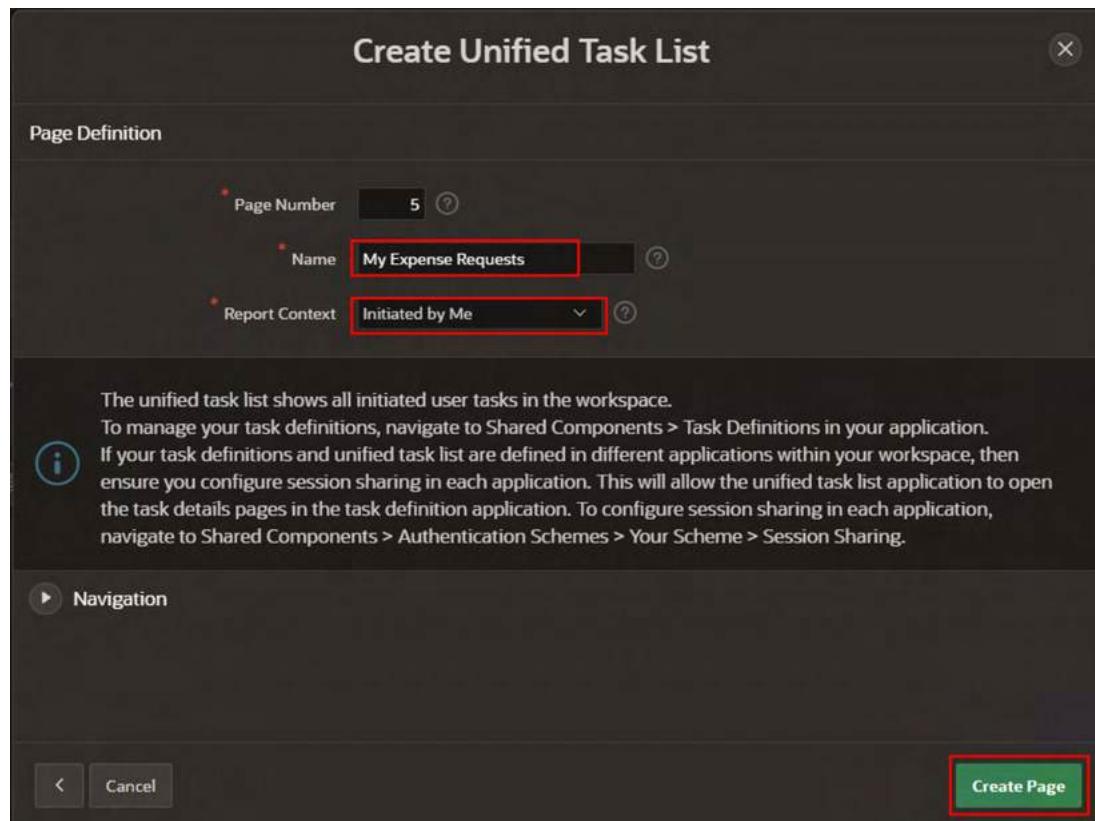
7. Under Component, select Unified Task List and click **Next**.



8. Specify the following page attributes:

- For Page Number - Type 5.
- For Name - Type **My Expense Requests**.
- For Report Context - Select **Initiated by Me**.

Click **Create Page**. A unified Task List page was created.



Create an Email Template

Add an Email Template to add for before expiry action in the task definition.

To define an email template:

1. Navigate to the Shared Components page:
 - On the Workspace home page, click **App Builder**.
 - Select an **Expense Tracker** application.
 - On the Application home page, click **Shared Components**.

- Under User Interface components, select Email Templates.

The screenshot shows the Oracle Application Builder's Shared Components page. The 'User Interface' section is active. Under 'Templates', the 'Email Templates' item is highlighted with a red box.

- On the Email Templates page, click **Create Email Template**. The Details page appears.

The screenshot shows the 'Email Templates' details page. The 'Create Email Template' button is highlighted with a red box.

- Under **Identification**:

- For Template Name - Enter **BEFORE EXPENSE EXPIRY EMAIL**.
- For Email Subject - Copy and paste the below text:

```
Expense Request FOR #APEX_TASK SUBJECT# Requires your review
```

*Note: For substitution strings with the **#STRINGNAME#** format. You can pass in values for these strings using the Placeholder Values dialog box for the Process in Page Designer or the APEX MAIL API.

- Under **HTML Format**:

- For Header - Copy the text below and paste it into the Header:

```
<b style="font-size: 24px;">My Approvals</b>
```

- For Body - Copy the text below and paste it into the Body:

```
<strong>Hello #APEX_TASK_OWNER#</strong>,<br>
    <br>Please check your "My Approvals" inbox. The expense
    request for #EMP_NAME# requires your timely review manner.<br>
        <br>Thanks for your kind attention to this matter.<br>
            <br>Need to make a change to your Approvals? <a
            href="#APPROVAL_URL#">Manage your Approvals here.</a>
```

- For Plain Text Format - Copy the text below and paste it into Plain text format:

```
Hello #APEX_TASK_OWNER#,  

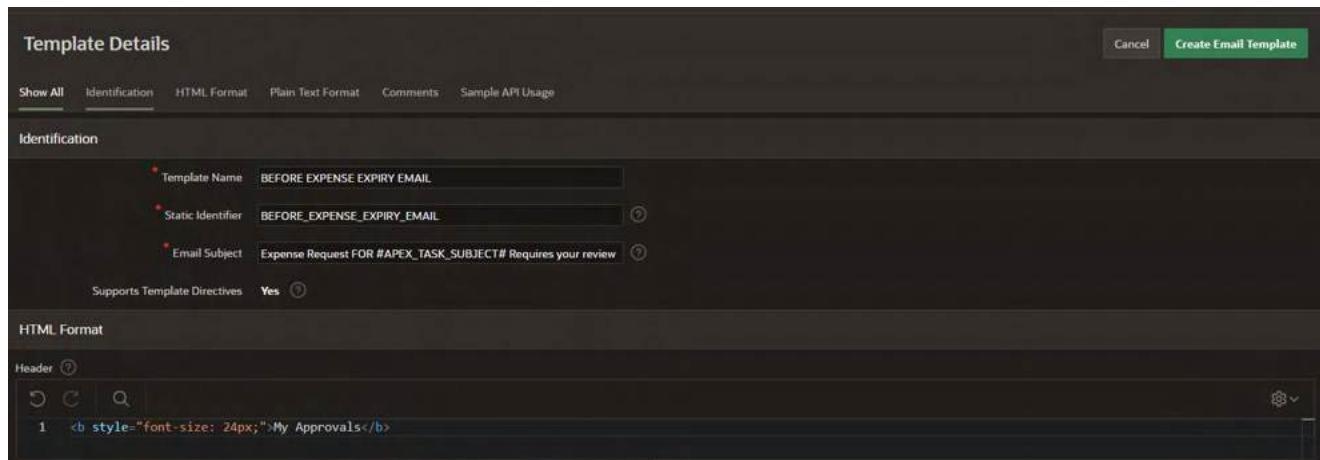
Please check your "My Approvals" inbox. The expense request for  

#EMP_NAME# requires your timely review manner.  

Thanks for your kind attention to this matter.  

Need to make a change to your Approval? Manage your Approval  

here: #APPROVAL_URL#
```



Template Details

Show All Identification HTML Format Plain Text Format Comments Sample API Usage

Identification

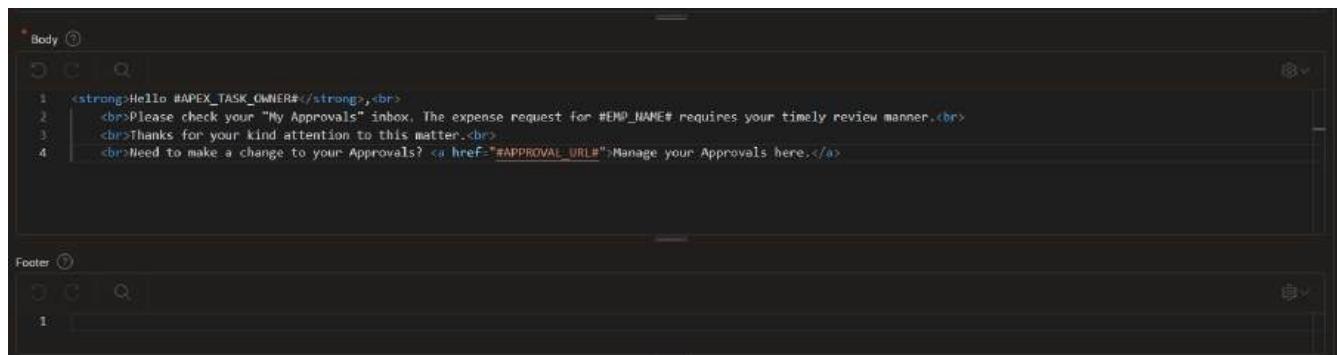
- Template Name: BEFORE EXPENSE EXPIRY EMAIL
- Static Identifier: BEFORE_EXPENSE_EXPIRY_EMAIL
- Email Subject: Expense Request FOR #APEX_TASK_SUBJECT# Requires your review

Supports Template Directives: Yes

HTML Format

Header:

```
1 <b style="font-size: 24px;">My Approvals</b>
```



Body

```
1 <strong>Hello #APEX_TASK_OWNER#</strong>,<br>
2     <br>Please check your "My Approvals" inbox. The expense request for #EMP_NAME# requires your timely review manner.<br>
3         <br>Thanks for your kind attention to this matter.<br>
4             <br>Need to make a change to your Approvals? <a href="#APPROVAL_URL#">Manage your Approvals here.</a>
```

Footer

```
1
```

The screenshot shows the Oracle APEX Script Editor interface. At the top, there are tabs: 'Show All', 'Identification', 'HTML Format', 'Plain Text Format' (which is selected), 'Comments', and 'Sample API Usage'. Below the tabs, the title 'Plain Text Format' is displayed. The main area contains a text editor with the following content:

```
1 Hello #APEX_TASK_OWNER#,
2 Please check your "My Approvals" inbox. The expense request for #EMP_NAME# requires your timely review manner.
3 Thanks for your kind attention to this matter.
4 Need to make a change to your Approval? Manage your Approval here: #APPROVAL_URL#
```

Below the text editor, there are two sections: 'Comments' and another 'Comments' section. The second 'Comments' section is currently empty.

6. Click Create Email Template.

Update Table Employee Details

1. Navigate to SQL Workshop and click SQL Commands.
2. Copy and paste the commands below into the Script Editor to update the Employee Details Table and execute the command one by one.

Note: The steps to create and populate the `EMPLOYEE_DETAILS` table are shared in TASK 2. Ensure that you have that table created and populated before running the commands below.

Add a new column HR_MGR (HR Manager) to the existing Employee table EMPLOYEE_DETAILS and update the existing Employee records as shown below:

```
insert into EMPLOYEE_DETAILS(empno, emp_name) values (50, 'SOPHIE');

insert into EMPLOYEE_DETAILS(empno, emp_name) values (60, 'ROBIN');

Alter table "EMPLOYEE_DETAILS" add "HR_MGR" VARCHAR2(10);

Update EMPLOYEE_DETAILS set HR_MGR = 'SOPHIE' where EMPNO in (10,20);

Update EMPLOYEE_DETAILS set HR_MGR = 'ROBIN' where EMPNO in (30,40);
```

Note: You can also create the two Users, SOPHIE and ROBIN, using the Manage Users And Groups menu option under Workspace Administration as done in TASK 3.

3. Click **Run**.

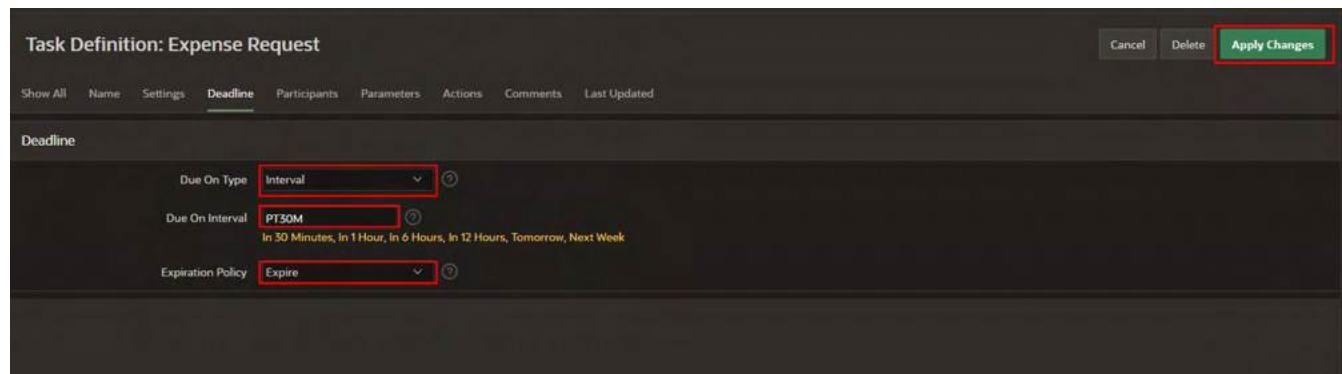
Update Task Definition

Add deadline and expiration events in actions for expense requests.

Navigate to App Builder and select Expense Tracker application. Click Shared Components > Workflows and Automations > Task Definitions and select the Expense Request Task Definition.

1. Under the **Deadline** section:

- For Due on type - Select interval.
- For Due on the interval - Type **PT30M**.
- For Expiration Policy - Select Expire.



2. Select **Expense Request**.

Under the **Actions** section - click **Add Actions**.

Specify the following attributes:

- For Name - Enter **BEFORE_EXPIRY**.
- For Type - Select Send Email.
- On Event - Select **Before Expire**.
- For Before Expire Interval - Enter **PT25M**.
- For Success Message - Enter **Task will expire in 5 minutes**.

Under the Send Email Settings section:

- For From - Enter the Email address of your wish.
- For To - Enter the Email address of your wish.
- For Email Template - Select **BEFORE EXPENSE EXPIRY EMAIL**.

The screenshot shows the 'Edit Action' interface for Oracle Application Express. The 'Action' tab is selected. The configuration details are as follows:

- Name: BEFORE_EXPIRY
- Type: Send E-Mail
- Execution Sequence: 40
- On Event: Before Expire
- Before Expire Interval: PT25M (1 hour before, 12 hours before, 24 hours before, 3 days before, 1 week before)
- Success Message: Task will expire in 5 minutes

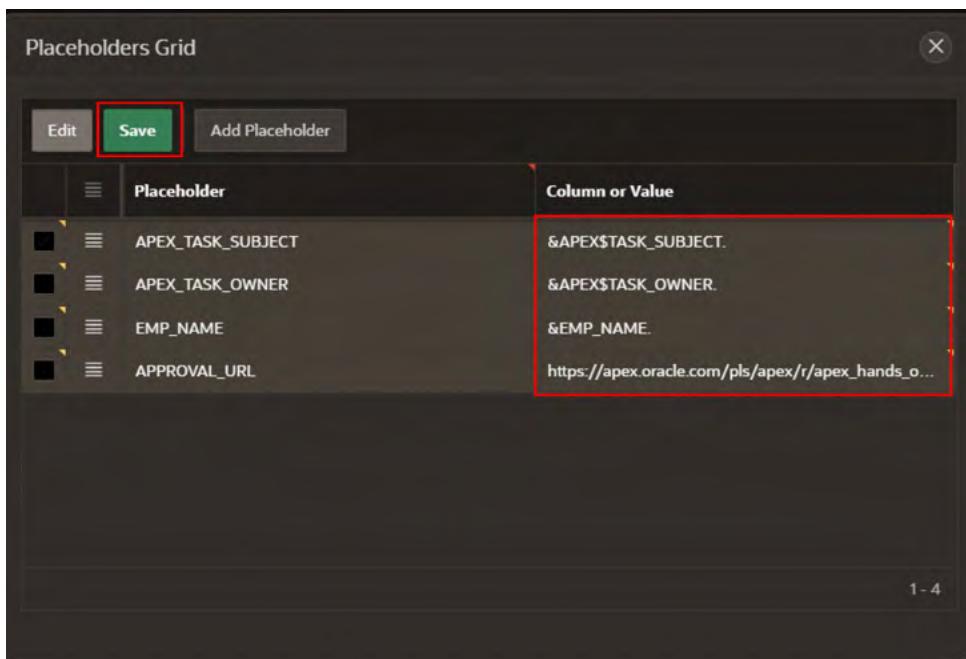
In the 'Send Email Settings' section, the 'From' field contains 'oracle-application-express_ww@oracle.com' and the 'To' field also contains 'oracle-application-express_ww@oracle.com'. The 'Create' button is visible in the top right corner.

- Click the **Set Placeholder Values** button beside the email template.

Add a Column or Value for mentioned Placeholders and Click **Save** to add placeholders.

Table 3: Extend Application Capabilities | Lab 1: Manage Approvals Component

| Placeholder | Column or Value |
|-------------------|---|
| APEX_TASK SUBJECT | &APEX\$TASK SUBJECT. |
| APEX_TASK OWNER | &APEX\$TASK OWNER. |
| EMP_NAME | &EMP_NAME. |
| APPROVAL_URL | Paste the Login URL of your Expense Tracker Application |

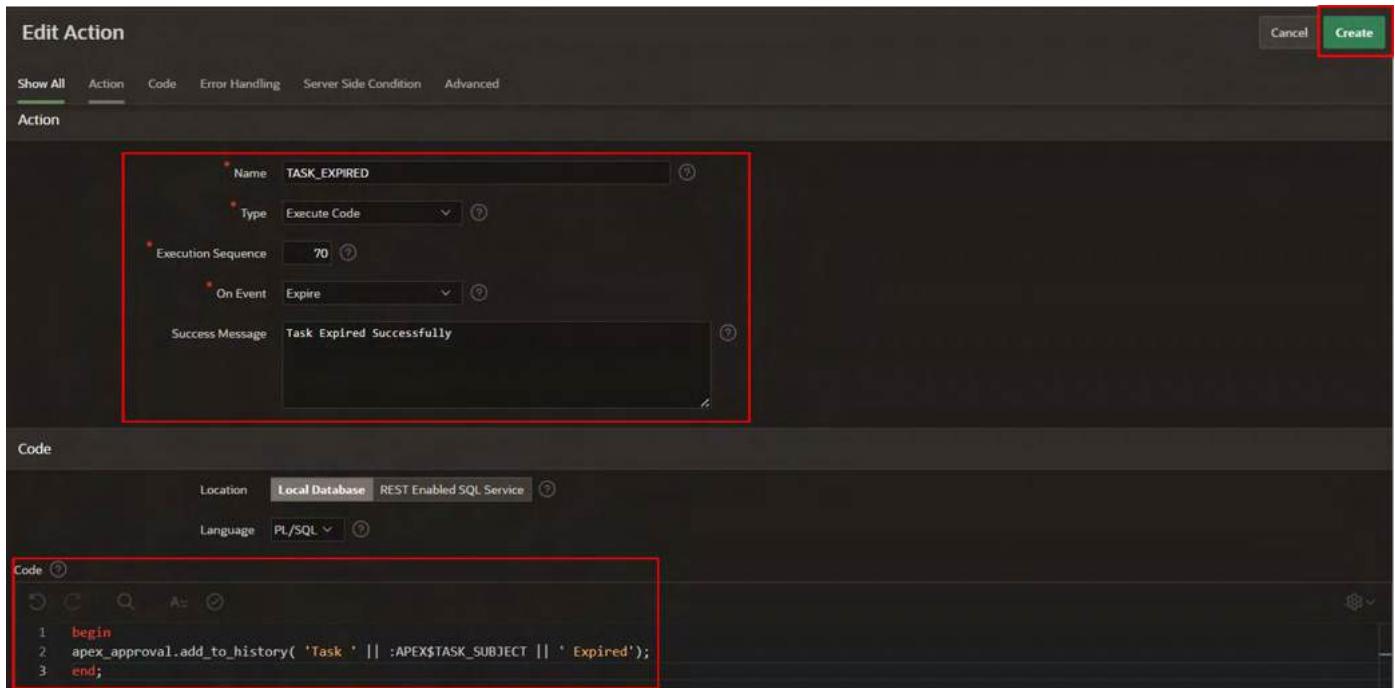


- Click **Create** to save an action.
- To add Expire event, click **Add Actions** and specify the following attributes:
 - For Name - Enter **TASK_EXPIRED**.
 - For Type - Select Execute Code.
 - For Execution Sequence - 70
 - On Event - Select **Expire**.

- For Success Message - Enter **Task Expired Successfully**.
- For Code: Copy the code below and paste it into the code editor:

```
begin
  apex_approval.add_to_history( 'Task ' || :APEX$TASK SUBJECT || '
  Expired');
end;
```

- Click **Create** to add action.



You now know how to manage Approval Components.

You may now **proceed to the next lab**.

Practice: Managing Application Data

Practice 1: Manage Application Data

Overview

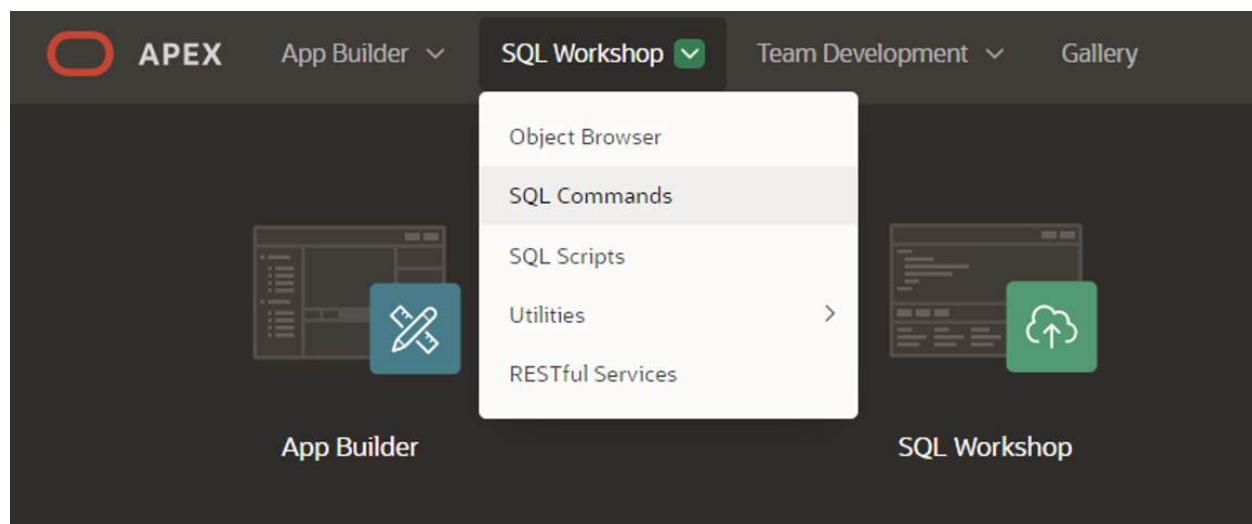
Oracle APEX allows you to build robust, feature-rich applications against remote, web, and REST data sources. In this lab, you log in to a remote database, create a database table, and then **register a schema with RESTful Services**. You will then create a **new application** and then consume the **REST Data Source** we created. Then, you will **synchronize** the data from REST Data Source to the Local Database table using **REST Data Synchronization**.

Enable Remote Database Schema to Oracle REST Data Services

Before starting this lab, you need to provision a new **APEX instance** or **Workspace**.

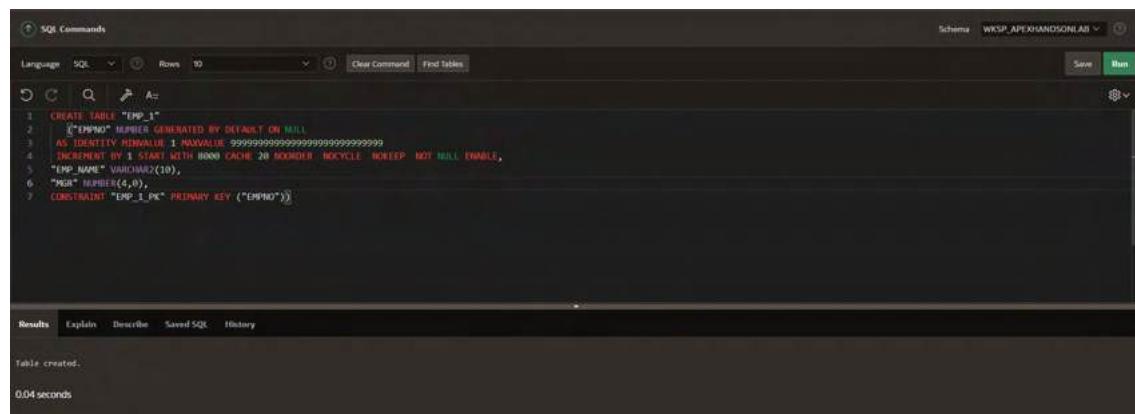
In this hands-on lab, you will log in to a new, remote APEX workspace, create a table, and then **register schema with ORDS**.

1. Log in to the new workspace you created.
2. Under **SQL Workshop**, select **SQL Commands**.

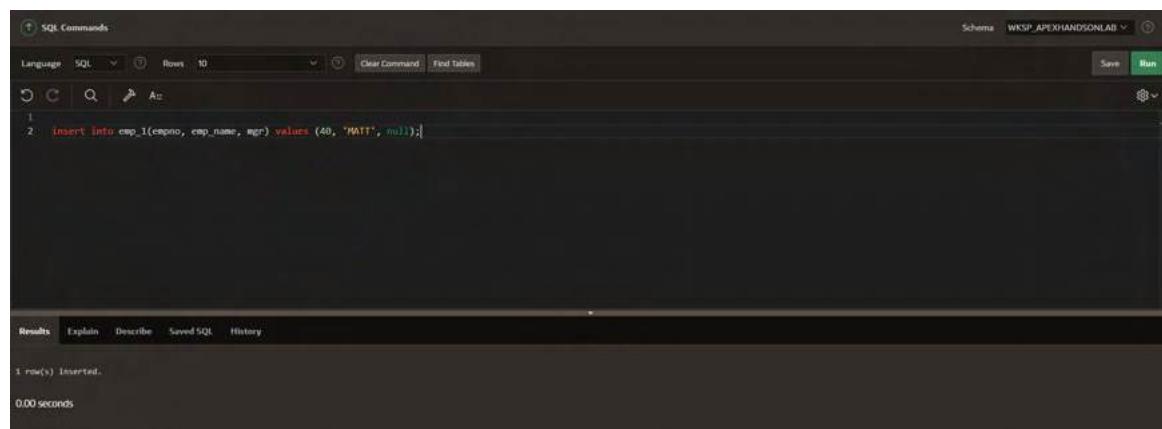


3. Now, copy the following statements and execute them one by one:

```
CREATE TABLE "EMP_1"
  ("EMPNO" NUMBER GENERATED BY DEFAULT ON NULL
   AS IDENTITY MINVALUE 1 MAXVALUE 99999999999999999999999999999999
   INCREMENT BY 1 START WITH 8000 CACHE 20 NOORDER NOCYCLE NOKEEP
   NOT NULL ENABLE,
  "EMP_NAME" VARCHAR2(10),
  "MGR" NUMBER(4,0),
  CONSTRAINT "EMP_1_PK" PRIMARY KEY ("EMPNO"))
insert into emp_1(empno, emp_name, mgr) values (10, 'JOHN', 30);
insert into emp_1(empno, emp_name, mgr) values (20, 'CLARA', 30);
insert into emp_1(empno, emp_name, mgr) values (30, 'JANE', 40);
insert into emp_1(empno, emp_name, mgr) values (40, 'MATT', null);
```

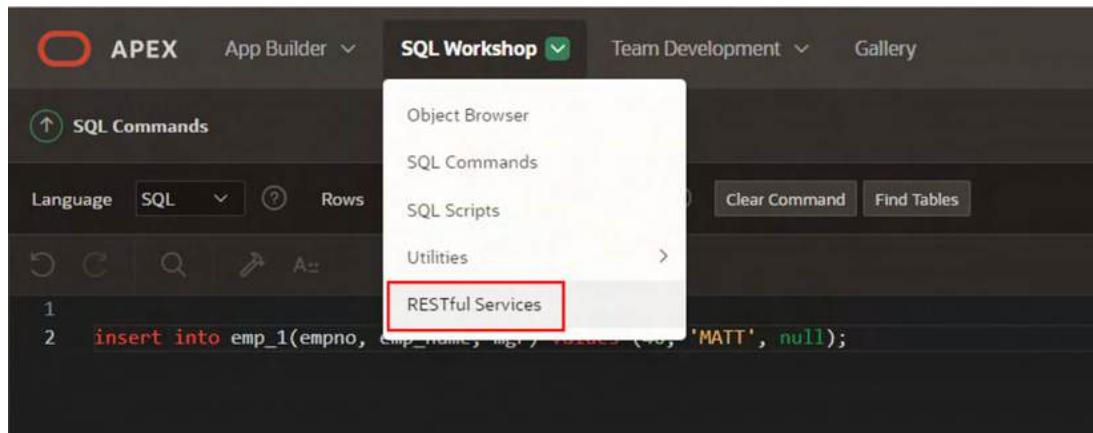


The screenshot shows the Oracle SQL Developer interface with the SQL Commands tab selected. The schema is set to 'WKSP_APEXHANDSONLAB'. The code area contains the CREATE TABLE statement for 'EMP_1' with its constraints and four insert statements. The results tab shows the message 'Table created.' and a execution time of '0.04 seconds'.

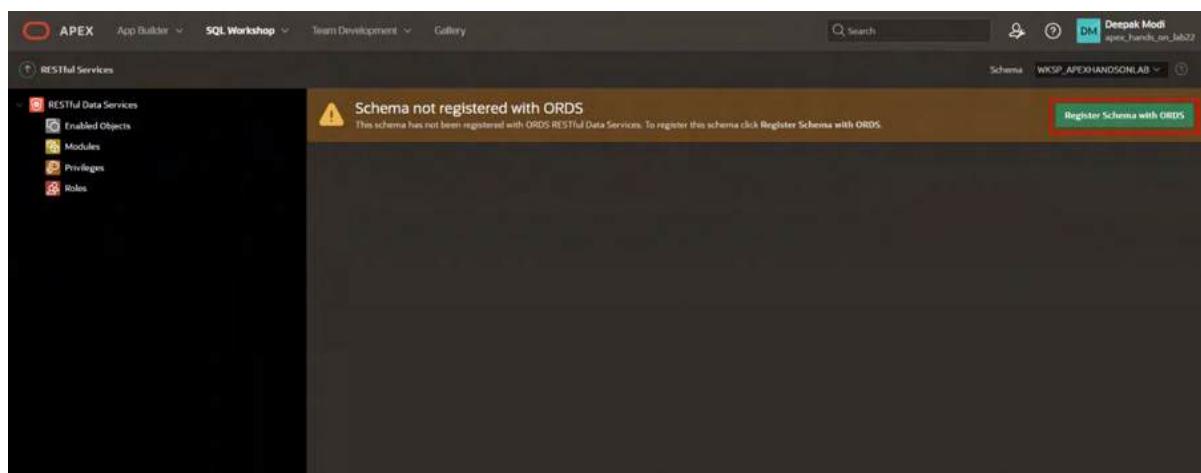


The screenshot shows the Oracle SQL Developer interface with the SQL Commands tab selected. The schema is set to 'WKSP_APEXHANDSONLAB'. The code area contains the first insert statement into 'EMP_1'. The results tab shows the message '1 row(s) inserted.' and a execution time of '0.00 seconds'.

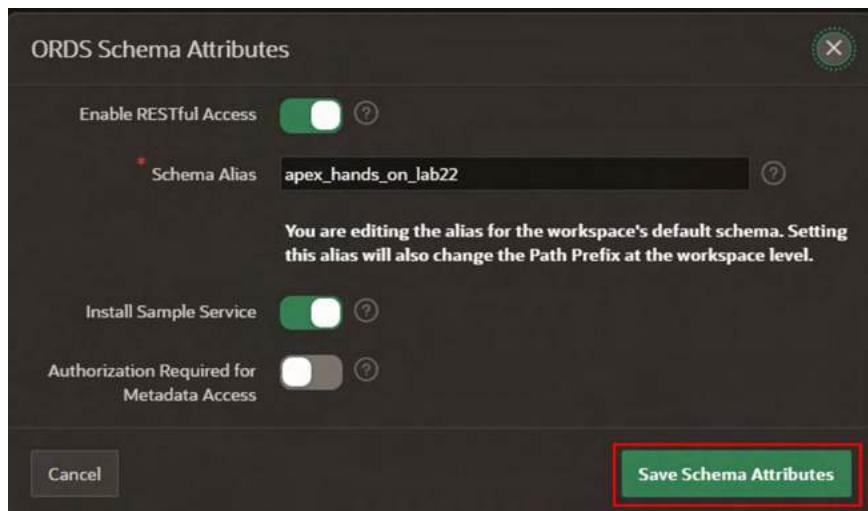
4. Select **RESTful Services** under **SQL Workshop**.



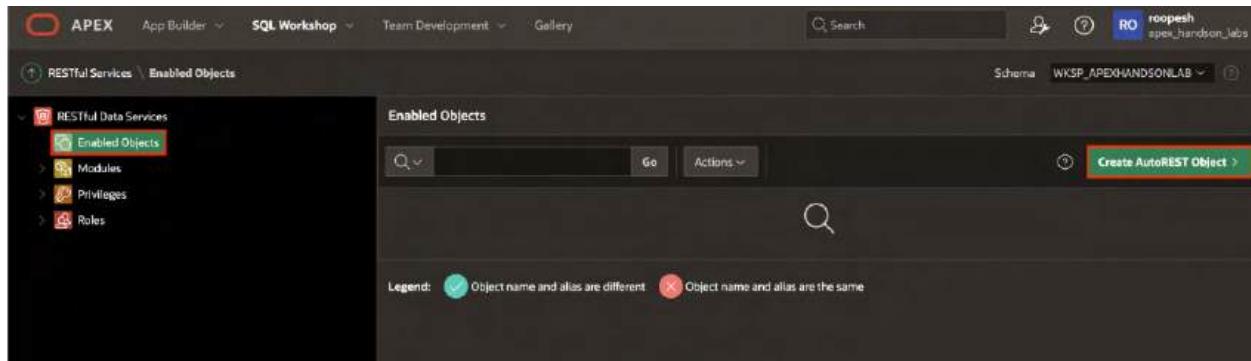
5. Click the **Register Schema with ORDS** button.



6. In the next window, leave the attributes to default and then click **Save Schema Attributes**.



7. Now, click **Enabled Objects** in the left Navigation Menu and then select **Create AutoREST Object** button.



8. Under **AutoREST Enable Object**, select the following and click **Create**.

- Set **Object Type** to **TABLE**.
- Select **EMP_1** for **Object**
- Copy the **Full URL** and paste it in your Note pad.

Then Click **Apply**.

A screenshot of the 'AutoREST Enable Object' dialog box. It has several input fields: 'Object Type' set to 'Table' (highlighted with a red box), 'Object' set to 'EMP_1' (highlighted with a red box), 'Object Alias' set to 'emp_1' (highlighted with a red box), 'Authorization Required' set to 'No' (highlighted with a red box), and a 'Full URL' field containing 'https://apex.oracle.com/ords/wksp_apexhandsontlabs/emp_1/' with a copy icon (highlighted with a red box). At the top right, there are 'Cancel' and 'Create' buttons, with 'Create' being highlighted with a red box.

✓ AutoREST definition created

Enabled Objects

| Parsing Object | Object Alias | Type | Status | AutoREST Authorization | Ops Allowed | Type Path | Pre Hook | Aliased |
|----------------|--------------|-------|---------|------------------------|-------------|-----------|----------|---------|
| EMP_1 | emp_1 | Table | ENABLED | DISABLED | | ENABLED | | |

Legend: Object name and alias are different Object name and alias are the same

- Now, please change the Copied URL and replace **ords** with **pls/apex** and Workspace Name with your Actual Workspace name as shown below.

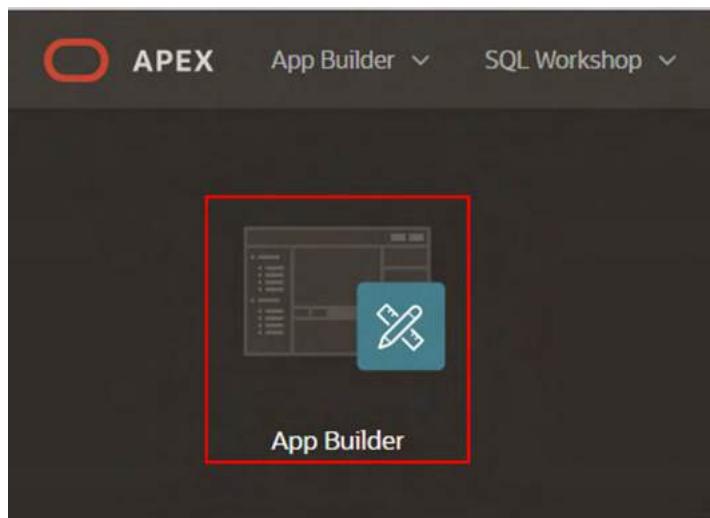
Copied URL: https://apex.oracle.com/ords/wksp_apexhandsonlabs/emp34/

Updated URL: https://apex.oracle.com/pls/apex/apex_handson_labs/emp_1/

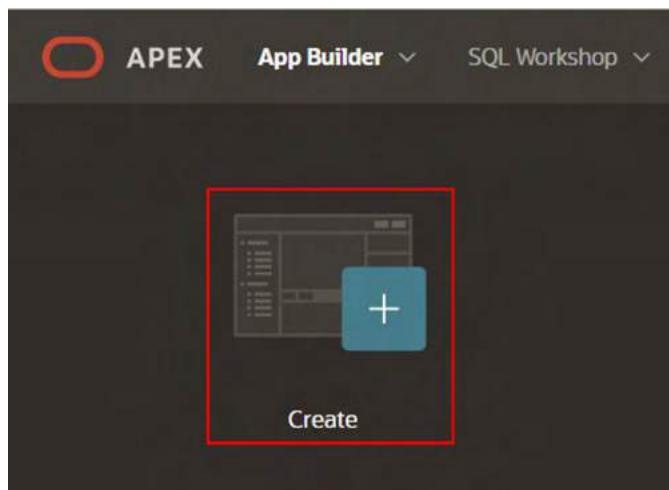
Create an Application and Integrate Application Data from REST Data Source

In this lab, you create a **New Application** and then consume the **REST Data Source** we created in the previous task. Then, you will **synchronize** the Data from REST Data Source to the Local Database table using **REST Data Synchronization**.

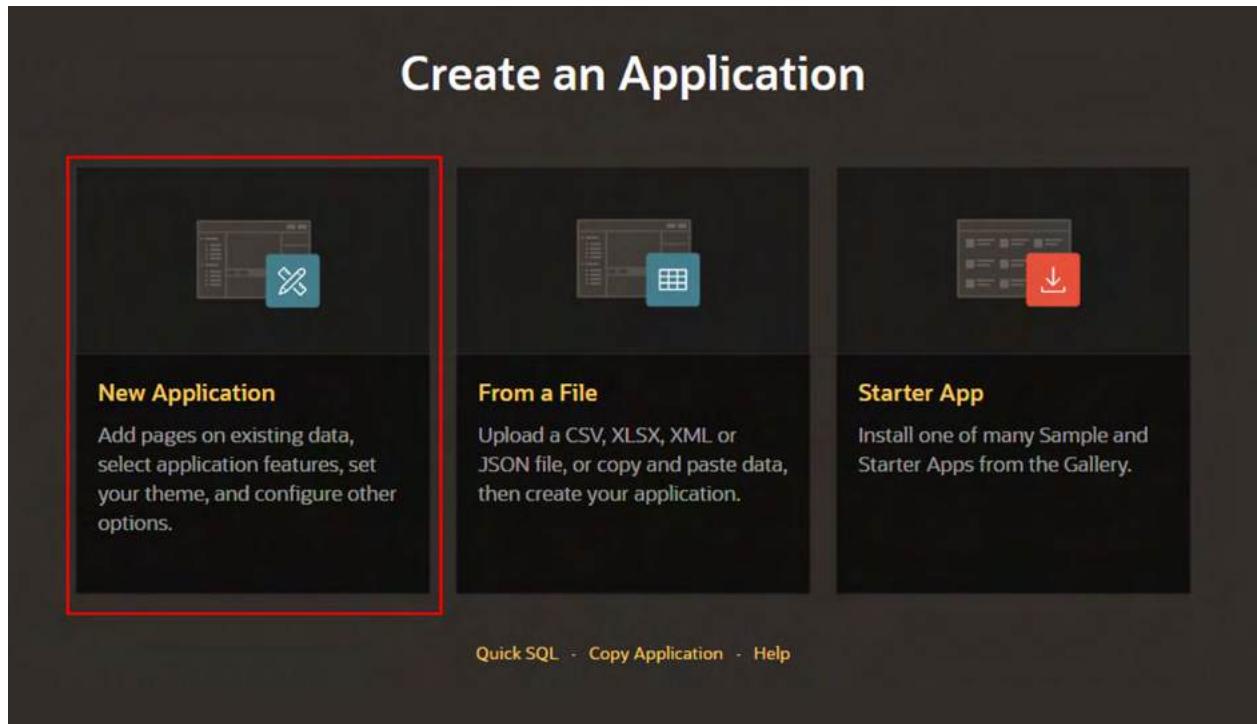
- Log in to the first workspace, where we have already created multiple applications.
- Select **App Builder** under Workspace Home Page.



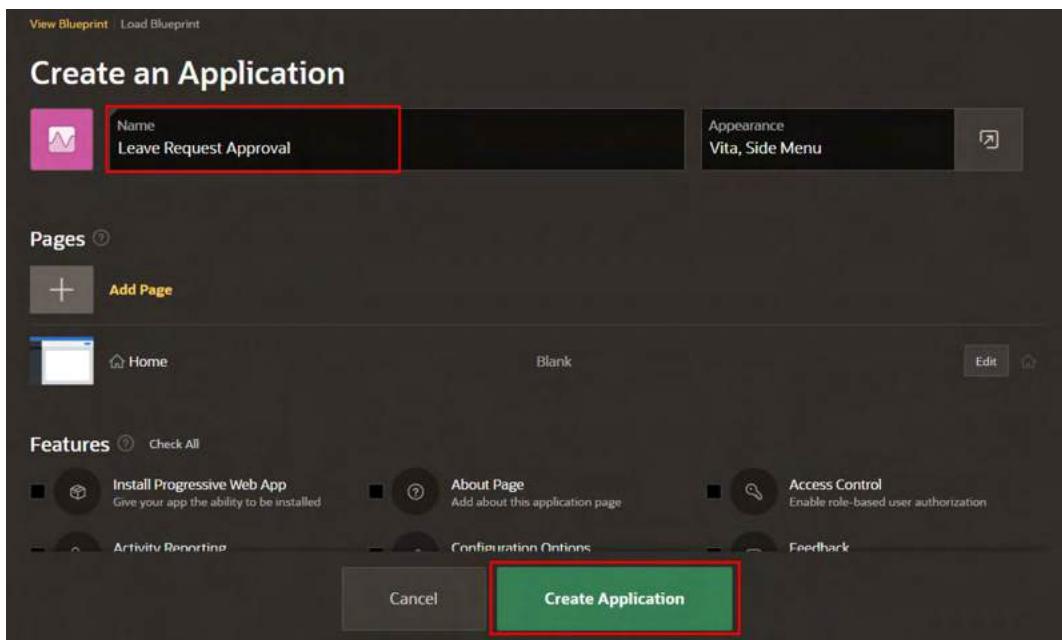
- Now click the **Create** button to create a new application.



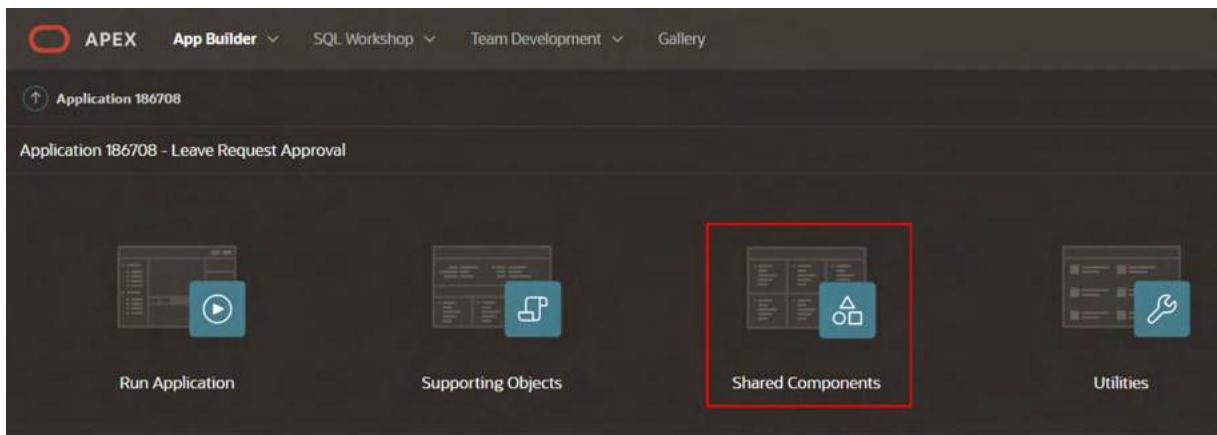
- Under **Create an Application**, select **New Application**.

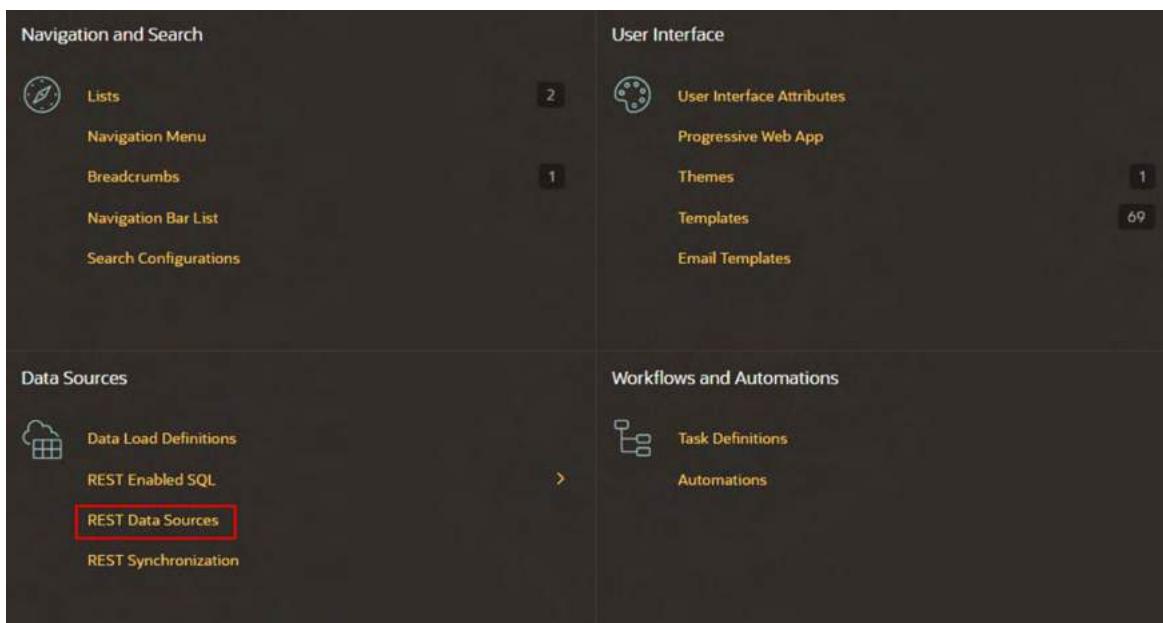


5. Enter the following and click **Create Application**.
- For Name - Enter Leave Request Approval.

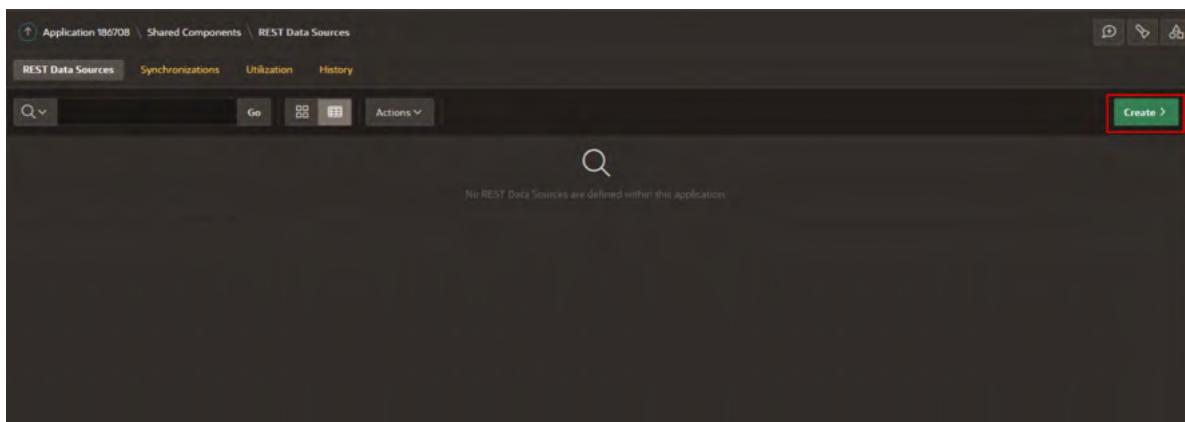


6. Navigate to **Shared Components** and then Select **REST Data Sources**.

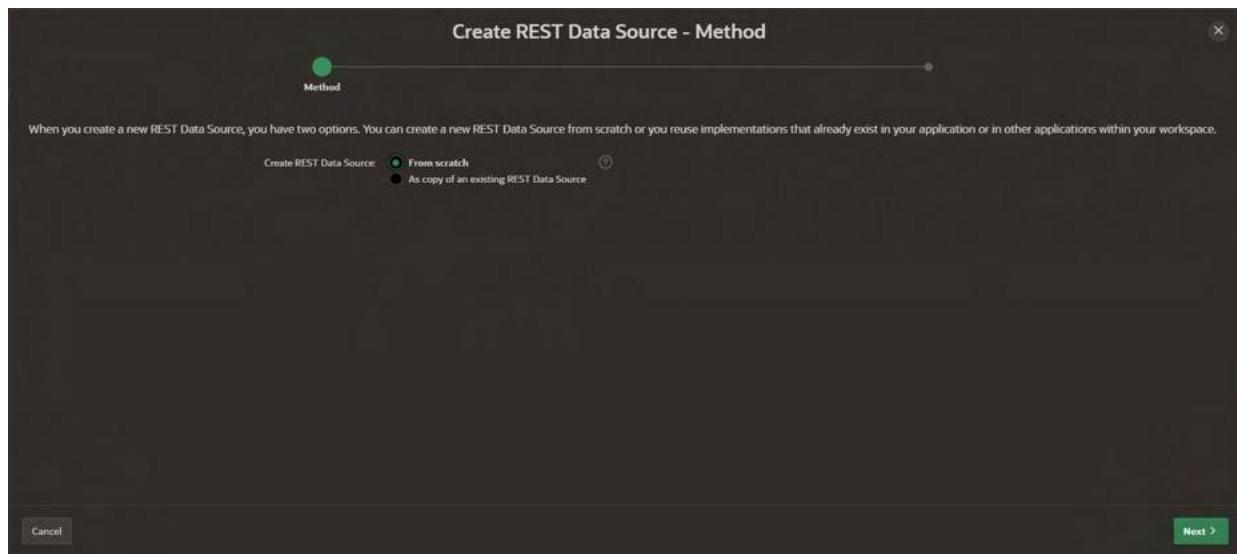




7. Under **REST Data Sources**, click the **Create** button.

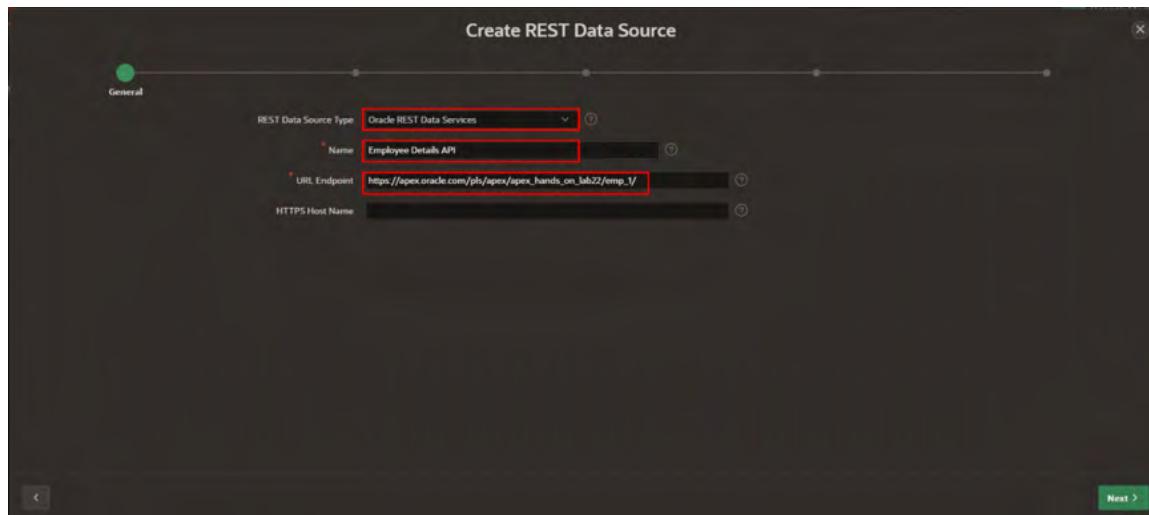


8. Under **Create REST Data Source - Method**, leave the settings to default and select **Next**.

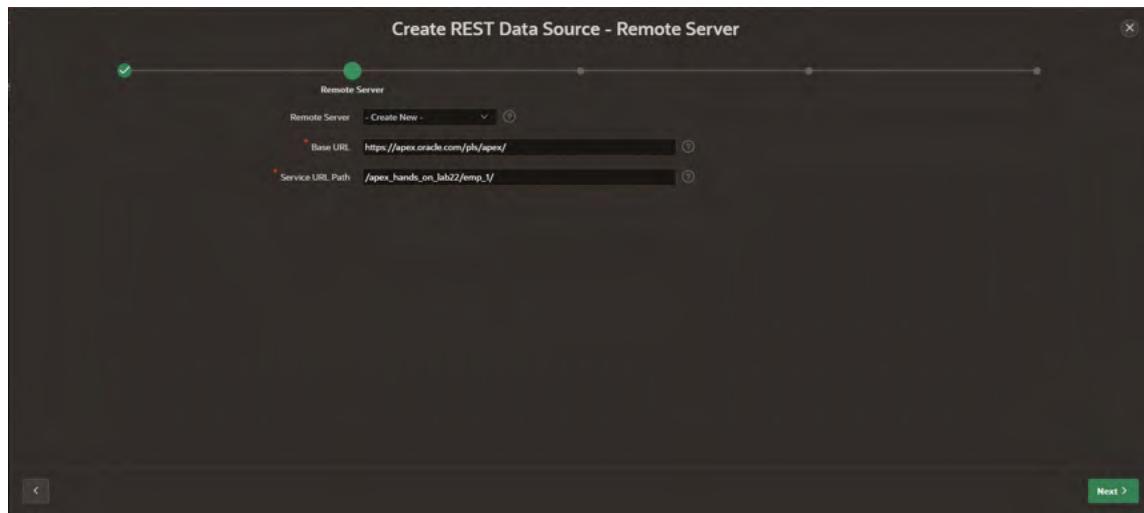


9. Under **Create REST Data Source**, enter the following and click **Next**.

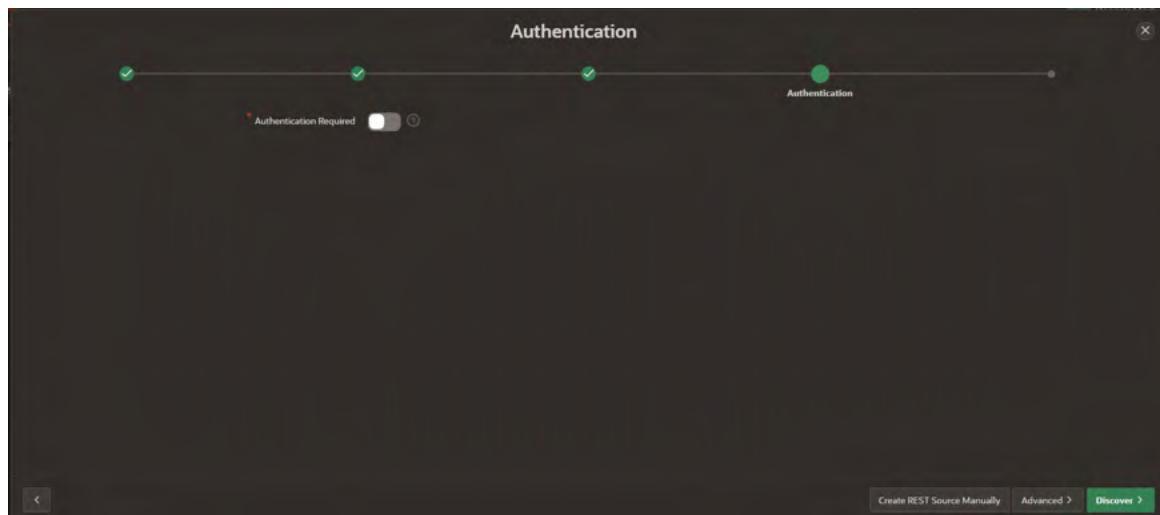
- For REST Data Source Type - select Oracle REST Data Services.
- For Name - Enter Employee Details API.
- For URL Endpoint - Enter the URI we copied in **Task 1**.



10. Now, under **Create REST Data Source - Remote Server**, leave all fields to **Default** and then click **Next**.



11. For Authentication, set **Authentication Required** to **No**. Click **Discover**.



12. Click the **Create REST Data Source** button.

The screenshot shows the 'REST Data Source Discovery' interface. At the top, there is a progress bar with five green checkmarks and a 'Preview' button. Below the progress bar, there are two tabs: 'Data' (selected) and 'Data Profile'. The 'Data' tab displays a table with four columns: 'Empno', 'Last Name', 'First Name', and 'Mgr'. The data rows are: 10, JOHN, SMITH, 20; 20, CLARA, LEE, 30; 30, JANE, DAVIS, 40; 40, MATT, WILSON, 50. At the bottom right of the table area, it says '1 - 4'. At the bottom of the screen, there are 'More Detail' and 'Create REST Data Source' buttons.

Synchronize Data from REST Data Sources to a Local Table

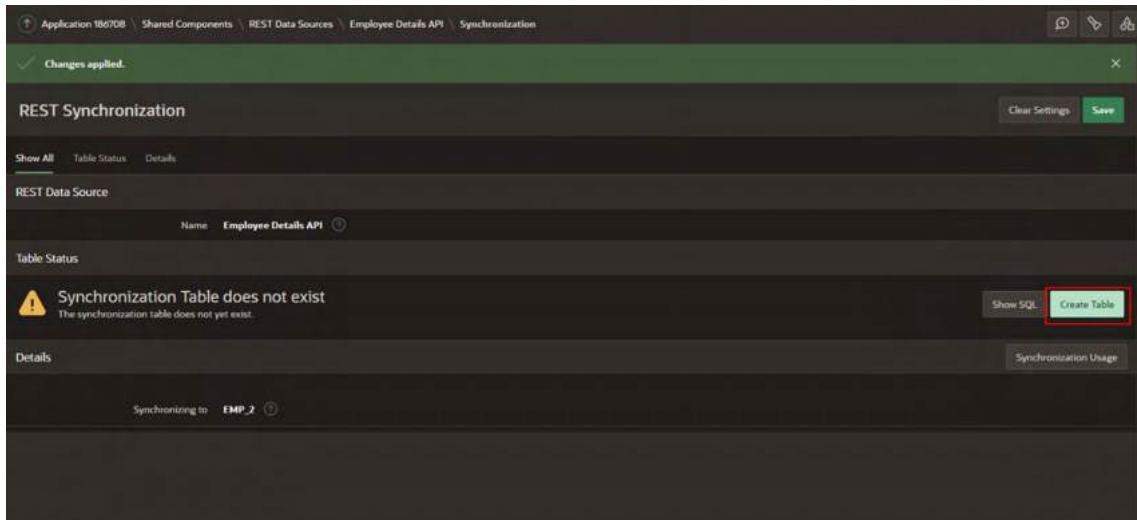
In this lab, you will **synchronize data** from the **REST Data source** you created in Task 2 to a **Local table**.

The **Data Synchronization** feature enables developers to automatically sync the contents of a local table with the data from an external REST service. Basically, APEX invokes the REST Service defined in the REST Data Source, downloads all data, and synchronizes it to a local table.

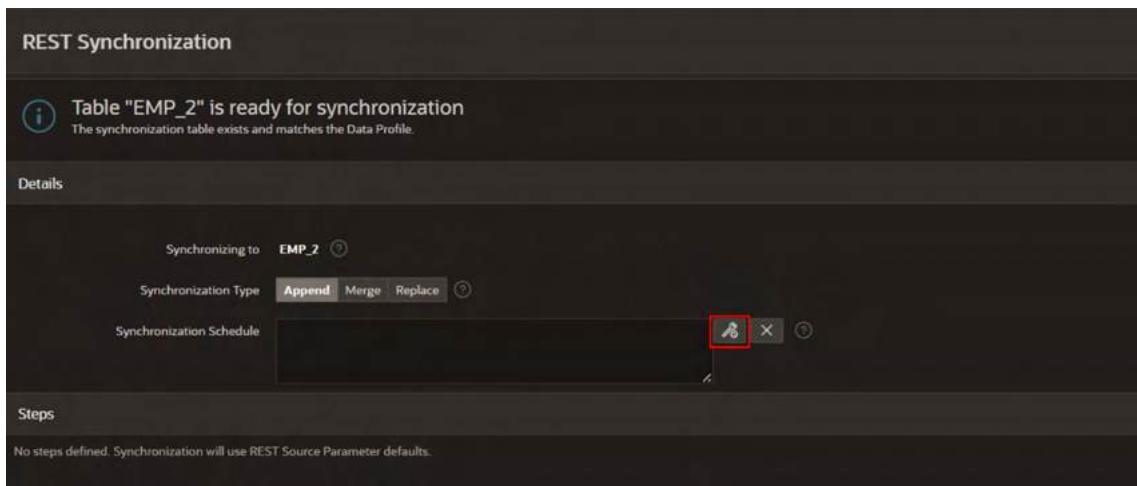
1. Now that you have created a **REST Data Source**, Under **REST Data Sources**, select **No** under the **Synchronized** column for **Employee Details API**.

The screenshot shows the 'REST Data Sources' page. At the top, there is a message 'REST Data Source created.' with a red border. Below the message, there are tabs: 'REST Data Sources' (selected), 'Synchronizations', 'Utilization', and 'History'. There is also a 'Copy' and 'Create >' button. The main table has columns: 'REST Source Name', 'Synchronized', 'Operations', 'Endpoint URL', 'Authentication', and 'Updated'. One row is shown: 'Employee Details API', 'No' (highlighted with a red border), 'S', 'https://apex.oracle.com/pls/apex/apex_hands_on_lab22/emp_1/', 'No', and '1 seconds ago'. At the bottom right of the table area, it says '1 - 1'.

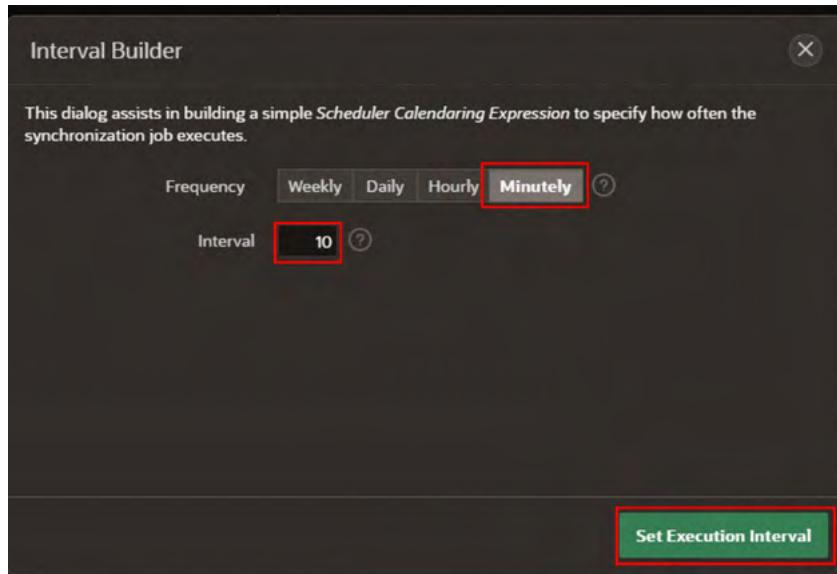
- Synchronization** is not configured yet. In this step, you provide the name of a new table or an existing table. In this example, select New Table for Synchronize To and enter **EMP_2** for Table Name. Then, click **Save**.
- The synchronization table is not created yet. You can view the SQL that is used to create the table. Click **Show SQL**.
- Under Table Status, click **Create Table**. The table **EMP_2** is now ready for synchronization. Notice the message **The synchronization table exists and matches the Data Profile**.



- You will define a **synchronization schedule** now. In the Synchronization Schedule field, you can use the Database Scheduler calendaring syntax to define repeating synchronization schedules, such as "every Tuesday and Friday at 4:00 p.m." or "the second Wednesday of every month." This calendaring syntax can be provided manually. Click the **Schedule Builder** button next to **Synchronization Schedule**.



6. The Interval Builder dialog box is displayed. In this example, for Frequency, select **Minutely** and enter 10 for Interval. Then, click **Set Execution Interval**.



7. You are now ready to test the data synchronization. Click **Save and Run**.



8. You can see that **data synchronization is triggered**.



You now know how to log in to a remote database, create a database table, and then **register a schema with RESTful Services**. Also, how to create a **new application**, consume the **REST Data Source** and **synchronize** the data from a REST Data Source to the Local Database table using **REST Data Synchronization**.

You may now **proceed to the next lab**.

Practice: Implement Security in your Application

Practice 1: Create Authentication and Authorization Schemes

Overview

Application security is very important for the majority of applications. You must ensure that users enter valid credentials. Generally, username and password (Authentication) and the logged-in user has appropriate rights within the application (Authorization).

Estimated Time: 20 minutes

Downloads

- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Implement Social Sign In(Facebook) Authentication in Online Shopping Cart Application.

In this hands-on lab, you create a Social Sign-in authentication scheme to enable Facebook Authentication. You test the authentication scheme.

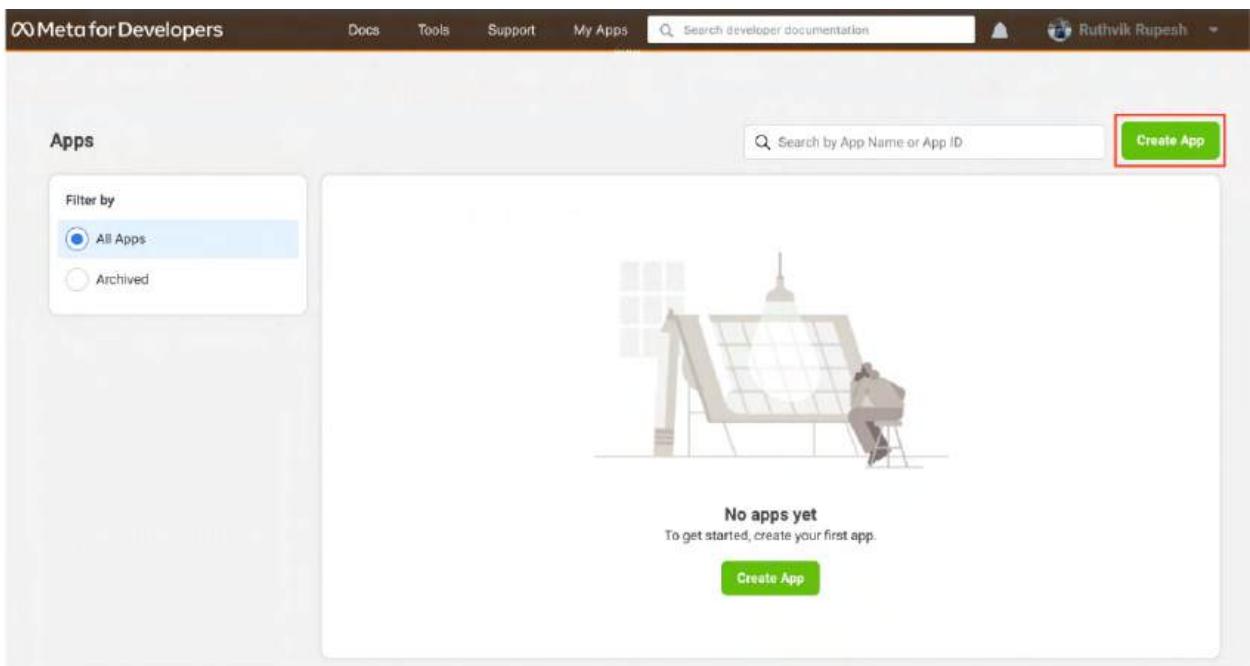
- Log in to Facebook's developer [console](#). Login to the Meta for Developers using your **Facebook Credentials** and then select **My Apps**.



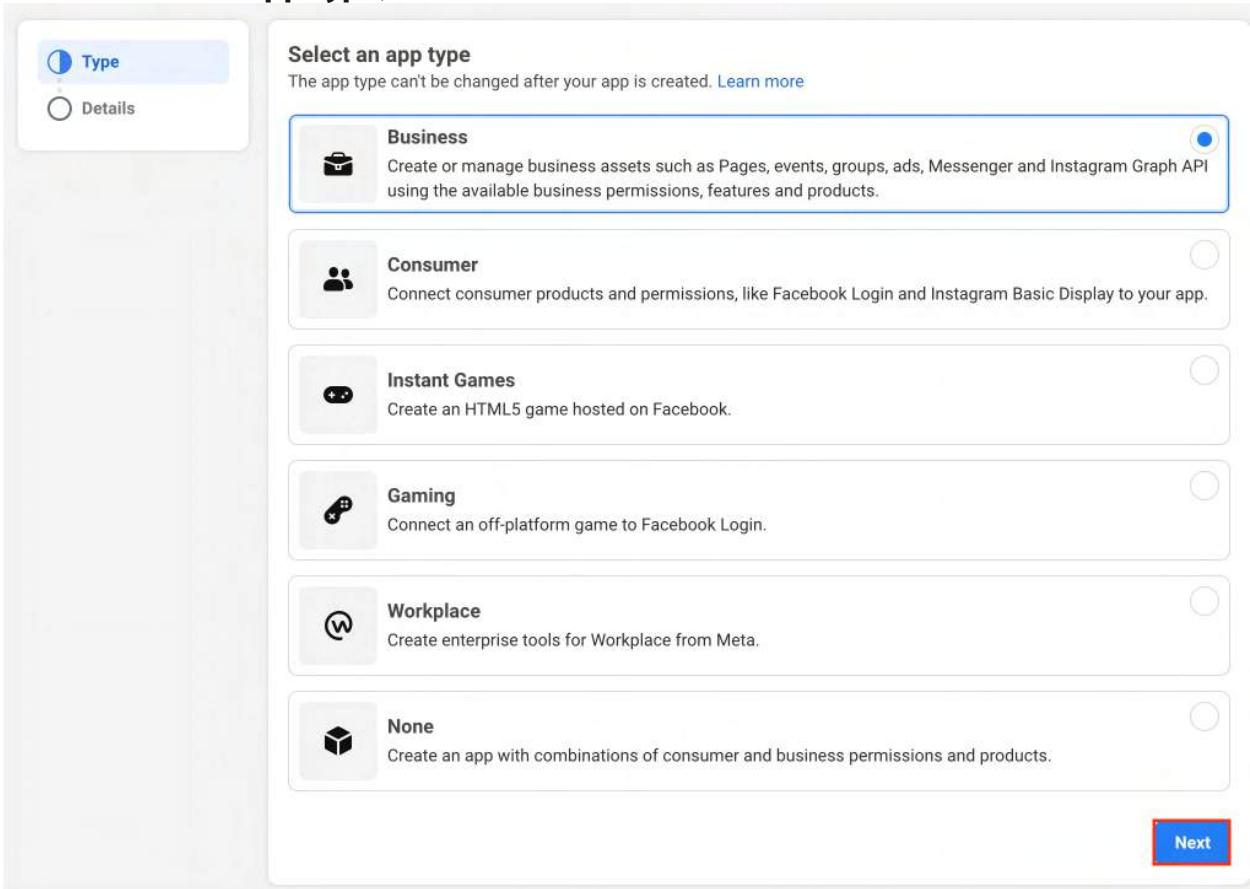
Note: If you are logging in to the Facebook developer console for the first time, you must follow the below process before navigating to My Apps

- A new user must first click the Get Started link to Create a Facebook for Developers account.*
- This then brings you to a welcome screen that gives you the option to Continue. Click Continue.*
- Review your email and agree or not to the marketing-related communication from Facebook.*
- Choose an option for who you are. Example, developer, product manager, etc.*

2. Click **Create App**.



3. Under **Select an app type**, Select **Business** and then click **Next**.



4. Now, Under **Provide Basic Information**, enter the following and click **Create App**.

- For **Display Name**, enter **APEX Authentication**
- For **App contact email**, enter **Your email address**.

Type

Details

Provide basic information

Display name
This is the app name associated with your app ID. You can change this later.

APEX Authentication

App contact email
This email address is used to contact you about potential policy violations, app restrictions or steps to recover the app if it's been deleted or compromised.

Business Account · Optional
To access certain permissions or features, apps need to be connected to a Business Account.

No Business Manager account selected

By proceeding, you agree to the [Facebook Platform Terms](#) and [Developer Policies](#).

Previous Create app

5. You will now verify your Facebook account by **Re-entering** your password.

Please Re-enter Your Password

For your security, you must re-enter your password to continue.

Password:

Forgot your password? Cancel Submit

6. Navigate to **Settings** and then select **Basic**. There, you can see the App ID and App Secret. Copy the **App ID** and **App secret** and paste them into your Notepad. It will be used to create the **web credential** in your **APEX application** later.

The screenshot shows the Facebook App Settings page. The left sidebar has links for Dashboard, Settings (with Basic selected), Advanced, Roles, Alerts, App Review, Products, and Activity Log. The main area shows the following details:

- App ID:** 3122217924703758
- App type:** Business
- Display name:** APEX Authentication
- Namespace:** (empty)
- App domains:** (empty)
- Contact email:** support@apexauthentication.com
- Privacy Policy URL:** Privacy policy for Login dialog and app details
- Terms of Service URL:** Terms of Service for Login dialog and App Details
- App icon (1024 x 1024):** (Placeholder image)
- Category:** Choose a category
- App purpose:**
 - This app's primary purpose is to access and use data from Facebook's Platform on behalf of:
 - Yourself or your own business
 - Clients

Define this option if the primary purpose of this app is to manage data or act on behalf of an individual client or multiple clients.

At the bottom right are **Discard** and **Save changes** buttons.

7. In the **Left Navigation Menu**, click on **Add Product**.

The screenshot shows the APEX Authentication application interface. At the top, there is a header bar with the text "APEX Authentication" and a dropdown arrow, followed by the text "App ID:" and a dropdown arrow. Below this is the left navigation menu. The menu items are:

- Dashboard
- Settings
 - Basic (highlighted with a blue background)
 - Advanced
- Roles
- Alerts
- App Review
- Products
 - [Add Product](#) (highlighted with a red border)
- Activity Log
- Activity Log

8. Under **Add products to your app**, select **Set up** in **Facebook Login** card.

The screenshot shows the 'Add products to your app' interface. It features a grid of six cards:

- App Events**: Understand how people engage with your business across apps, devices, platforms and websites. Buttons: Read Docs, Set up.
- Audience Network**: Monetize your app and grow revenue with ads from Meta advertisers. Buttons: Read Docs, Set up.
- Facebook Login**: The world's number one social login product. Buttons: Read Docs, Set up (highlighted with a red border).
- Fundraisers**: Create and manage fundraisers for charities. Buttons: Read Docs, Set up.
- Instagram Graph API**: Integrate your app with the Instagram API to let businesses use your app with their Instagram accounts. Buttons: Read Docs, Set up.
- Jobs**: Post jobs to the Facebook platform and receive applications from users. Buttons: Read Docs, Set up.

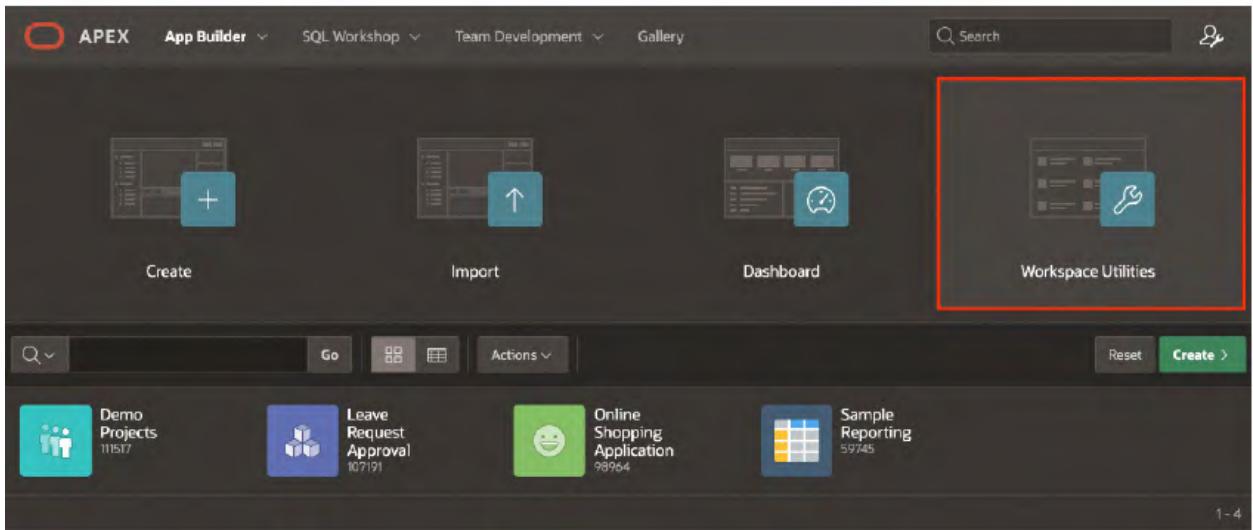
9. Navigate to **Settings** under **Facebook Login** in the navigation menu. You have to enable Client OAuth Login and add https://apex.oracle.com/pls/apex/apex_authentication.callback (If you are using apex.oracle.com) Valid OAuth Redirect URIs. Then click on **Save Changes**.

The screenshot shows the 'Client OAuth settings' page under the 'Facebook Login' section. The left sidebar shows the navigation menu with 'Facebook Login' selected. The main area displays several configuration options:

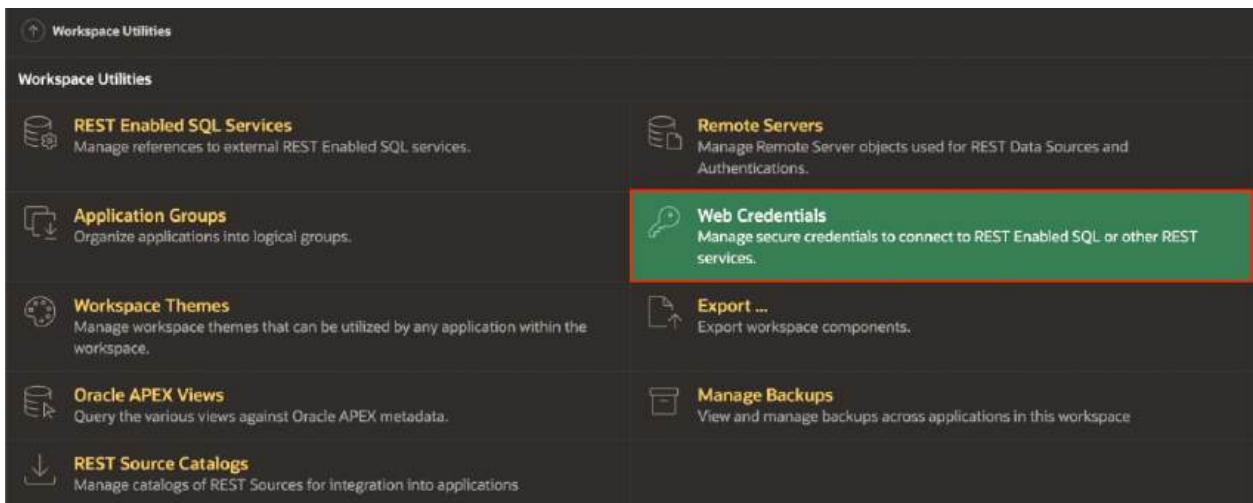
- Client OAuth login**: Enabled (Yes). Description: Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. (i)
- Web OAuth login**: Enabled (Yes). Description: Enables web-based Client OAuth Login. (i)
- Enforce HTTPS**: Enabled (Yes). Description: Enforce the use of HTTPS for Redirect URIs and the JavaScript SDK. Strongly recommended. (i)
- Force Web OAuth reauthentication**: Disabled (No). Description: When on, prompts people to enter their Facebook password in order to log in on the web. (i)
- Embedded Browser OAuth Login**: Disabled (No). Description: Enable webview Redirect URIs for Client OAuth Login. (i)
- Use Strict Mode for redirect URIs**: Enabled (Yes). Description: Only allow redirects that exactly match the Valid OAuth Redirect URIs. Strongly recommended. (i)
- Valid OAuth Redirect URIs**: A text input field containing the URL https://apex.oracle.com/pls/apex/apex_authentication.callback, which is highlighted with a red border.
- Login from Devices**: Enabled (Yes). Description: Enables the OAuth redirect flow for devices like a mobile phone.
- Login with the JavaScript SDK**: Enabled (Yes). Description: Enables login and授权 with the JavaScript SDK.

At the bottom right are 'Discard' and 'Save changes' buttons.

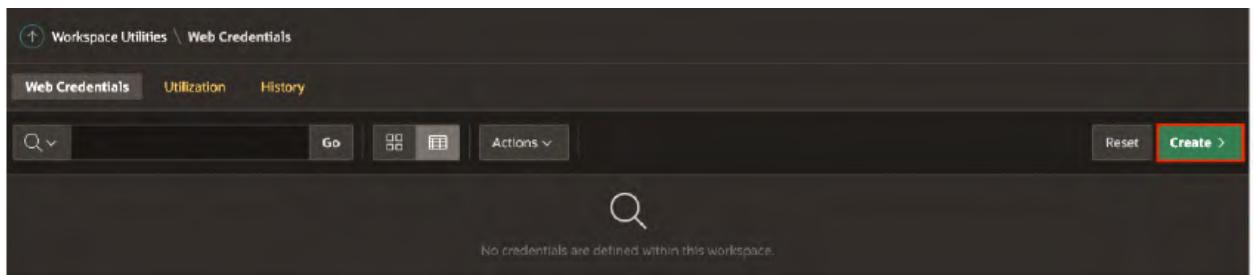
10. Login to your APEX workspace and click **Workspace Utilities**.



11. Under **Workspace Utilities**, Select **Web Credentials**.



12. Click **Create**.



13. In the **Web Credentials** enter the following and click **Create**. Under **Attributes**:

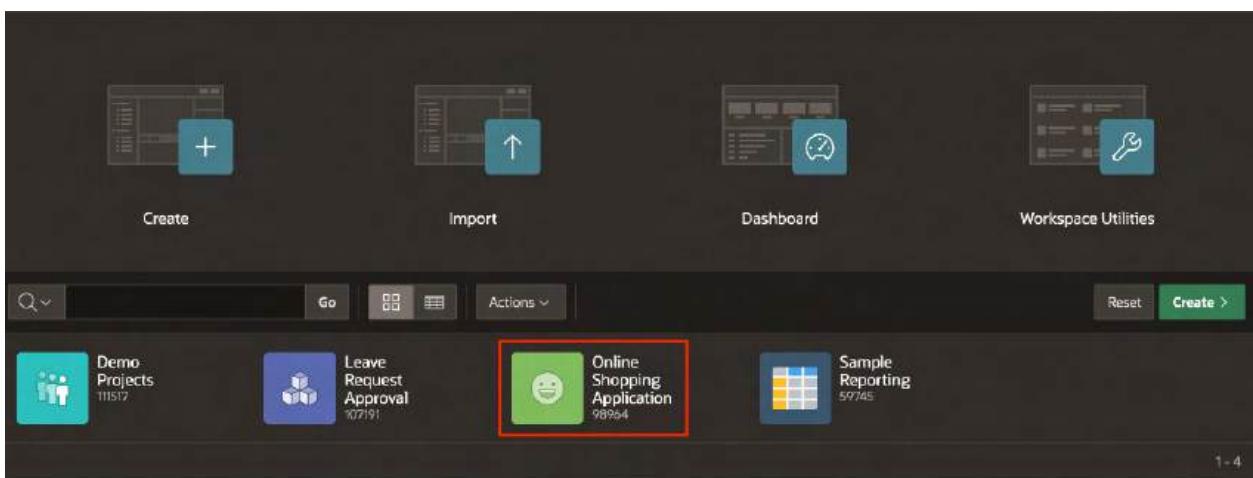
- For **Name**, Enter **FB_LOGIN_DEMO**
- For **Static Identifier**, Enter **FB_LOGIN_DEMO**
- For **Authentication Type**, select **OAuth2 Client Credentials Flow**.
- For **Client ID or Username**, Enter the **APP ID** you copied in **Step 6**.
- For **Client Secret or Password** and **Verify Client Secret or Password**, Enter the **App Secret** you copied in **Step 6**.

The screenshot shows the 'Web Credentials' creation screen in the Oracle Database workspace. The 'Attributes' section contains the following fields:

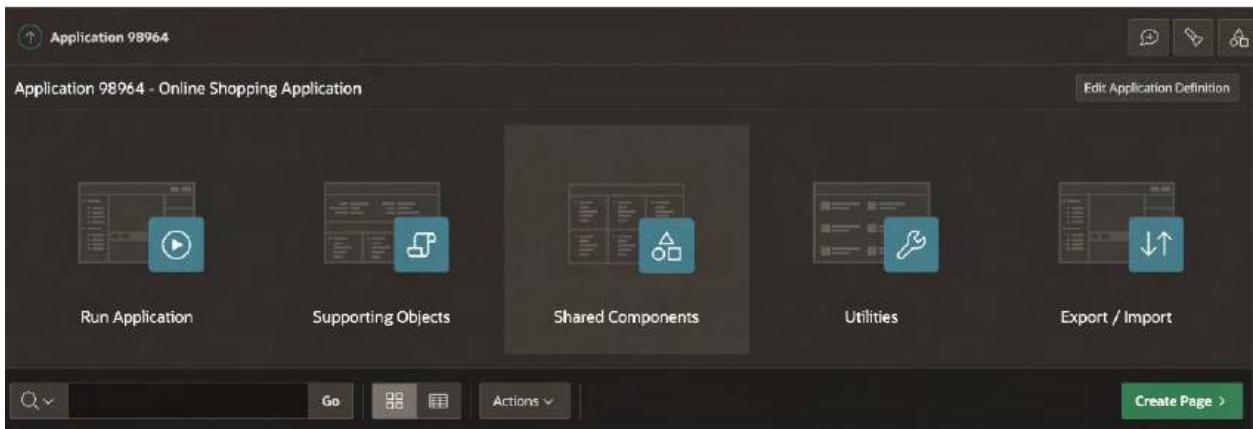
- Name: FB_LOGIN_DEMO
- Static Identifier: FB_LOGIN_DEMO
- Authentication Type: OAuth2 Client Credentials Flow
- Client ID or Username: 312217924703758
- Client Secret or Password: (Masked)
- Verify Client Secret or Password: (Masked)

Other visible fields include 'Valid for URLs' (empty), 'Prompt On Install' (set to 'On'), and a large 'Comments' text area. The 'Create' button is located at the top right of the form.

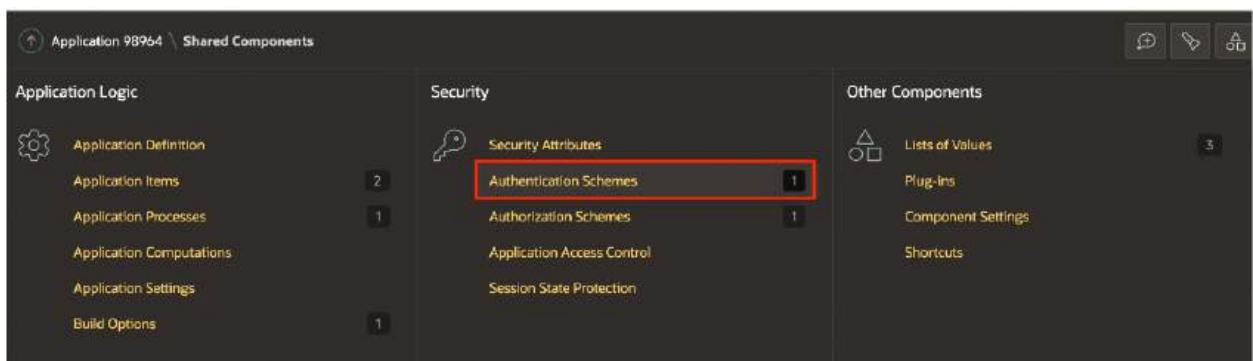
14. Navigate to **App Builder** and select **Online Shopping Application**.



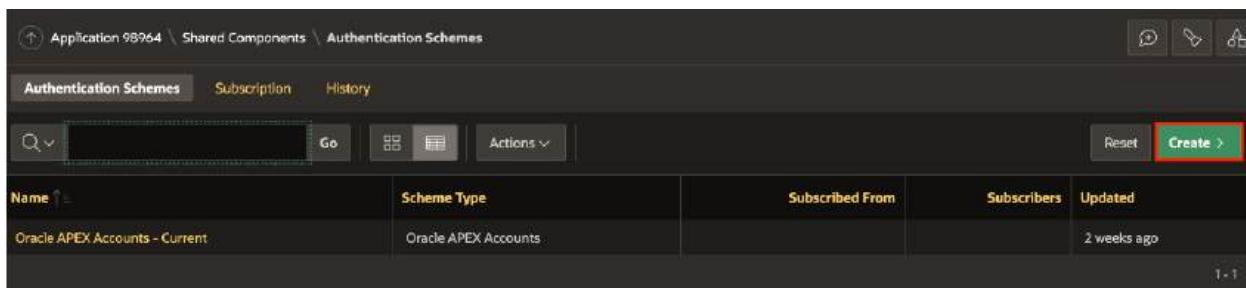
15. Click **Shared Components**.



16. Under **Security**, Select **Authentication Schemes**.

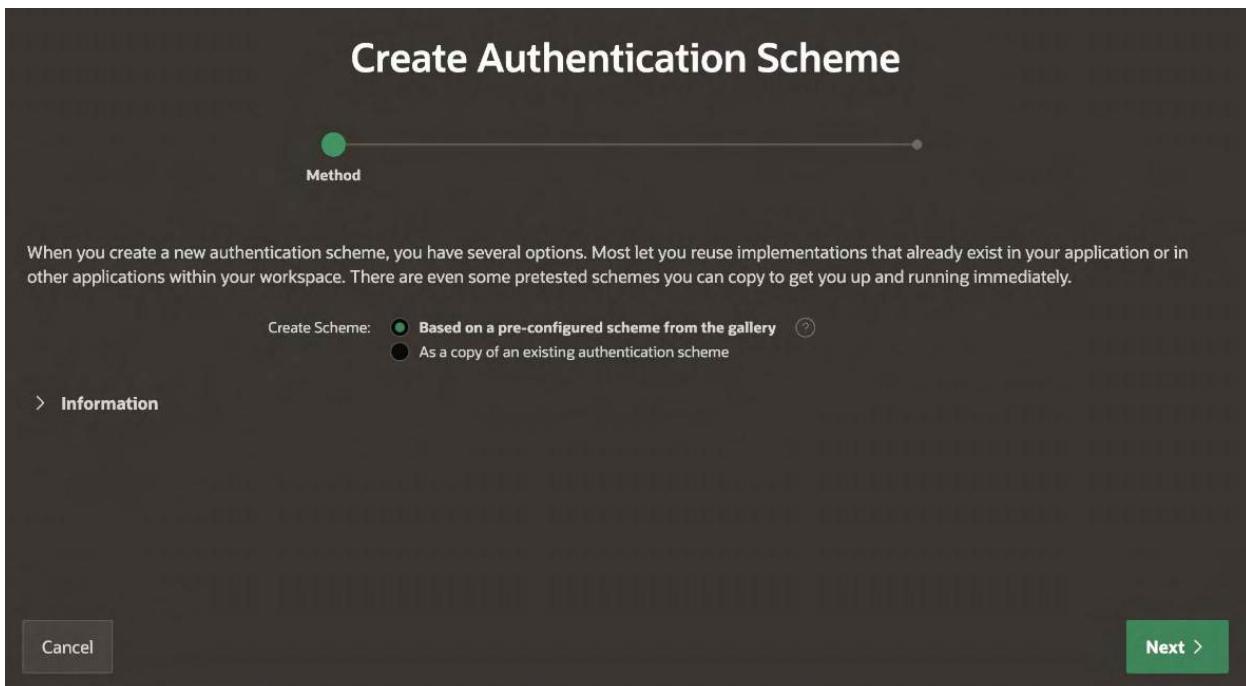


17. In the **Authentication Schemes** page, click **Create**.



The screenshot shows the 'Authentication Schemes' page in the Oracle Application workspace. The top navigation bar includes 'Application 98964', 'Shared Components', and 'Authentication Schemes'. Below the navigation is a search bar, a 'Go' button, and a toolbar with icons for search, refresh, and actions. A 'Reset' button and a 'Create >' button are on the right. The main area displays a table with columns: Name, Scheme Type, Subscribed From, Subscribers, and Updated. One row is visible: 'Oracle APEX Accounts - Current' (Scheme Type: Oracle APEX Accounts), Subscribed From: Oracle APEX Accounts, Subscribers: 1, Updated: 2 weeks ago.

18. Under **Create Authentication Scheme** Page, leave the settings to default and click **Next**.



The screenshot shows the 'Create Authentication Scheme' wizard. The title is 'Create Authentication Scheme'. Step 1 is 'Method'. It says: 'When you create a new authentication scheme, you have several options. Most let you reuse implementations that already exist in your application or in other applications within your workspace. There are even some pretested schemes you can copy to get you up and running immediately.' Below this are two options: 'Create Scheme: Based on a pre-configured scheme from the gallery' and ' As a copy of an existing authentication scheme'. At the bottom are 'Cancel' and 'Next >' buttons.

19. In the **Authentication Scheme** enter the following and click **Create Authentication Scheme**. Under **Name**:

- For **Name**, Enter **FB Authentication**.
- For **Scheme Type**, select **Social Sign-In**.

Under **Settings**:

- For **Credential Store**, Enter **FB_LOGIN_DEMO**.
- For **Authentication Provider**, select **Facebook**.
- For **Scope**, Enter **email**.

- For **Username**, Enter **name**.

Authentication Scheme

Name

- Name: FB Authentication
- Scheme Type: Social Sign-In

Subscription

Reference Master Authentication Scheme From: [dropdown] Refresh

This is the "master" copy of this authentication scheme.

There are no subscribers to this authentication scheme.

Settings

- Credential Store: FB_LOGIN_DEMO
- Authentication Provider: Facebook
- Scope: email
- Username: name
- Convert Username To Upper Case: No
- Additional User Attributes: [empty]
- Map Additional User Attributes To: [empty]
- Verify Attributes: Yes

20. Notice that a new **Authentication Scheme** you created is displayed as **FB Authentication - Current**.

Application 98964 \ Shared Components \ Authentication Schemes

Authentication Schemes Subscription History

Q Go Actions ▾ Reset Create >

| Name | Scheme Type | Subscribed From | Subscribers | Updated |
|-----------------------------|----------------------|-----------------|-------------|----------------|
| FB Authentication - Current | Social Sign-In | | | 95 seconds ago |
| Oracle APEX Accounts | Oracle APEX Accounts | | | 2 weeks ago |

21. Run the application by navigating to **Online Shopping Application** and click **Run**.

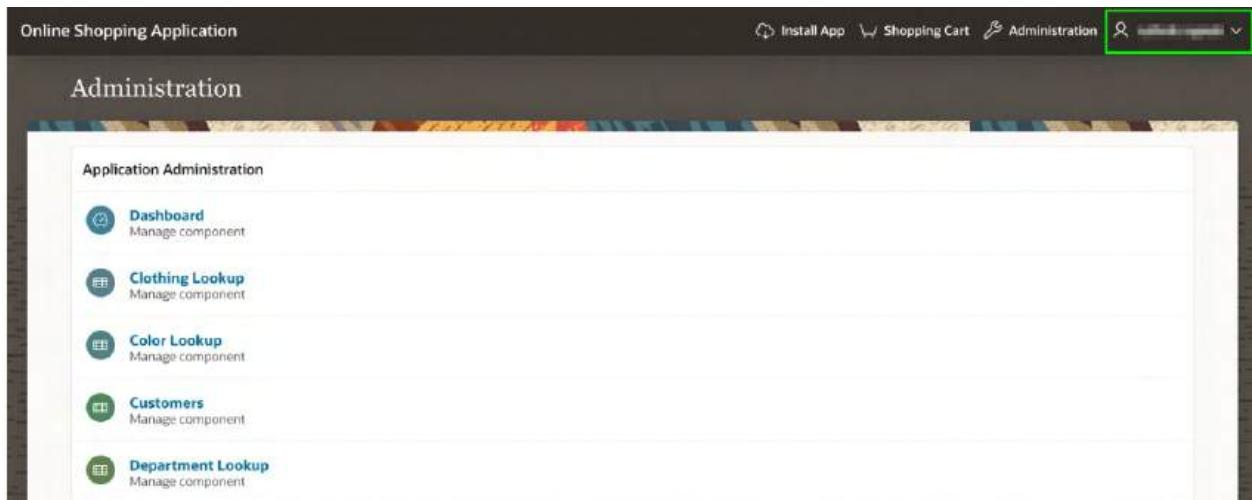
22. In the User Interface, click **Administration**.

The screenshot shows the 'Online Shopping Application' interface. At the top, there are links for 'Install App', 'Shopping Cart', 'Administration' (which is highlighted with a red border), and a user profile 'nobody'. On the left, a sidebar contains a search bar and filters for 'Color' (black, blue, grey, red, brown, green, white) and 'Department' (Boy's, Women's, Girl's, Men's). The main area displays a grid of products under the heading 'Total Products 46'. The first two products shown are 'Boy's Coat (Blue)' and 'Boy's Coat (Brown)', both priced at \$10.24.

23. Now, log in to the application using your **Facebook Credentials**.

The screenshot shows the Facebook login page. The top navigation bar includes the 'facebook' logo and a 'Create New Account' button. The main form is titled 'Log in to Facebook' and contains fields for 'Email or Phone' and 'Password', followed by a 'Log In' button. Below the form are links for 'Forgotten account?' and 'Create New Account'. At the bottom, there is a link for 'Not now'.

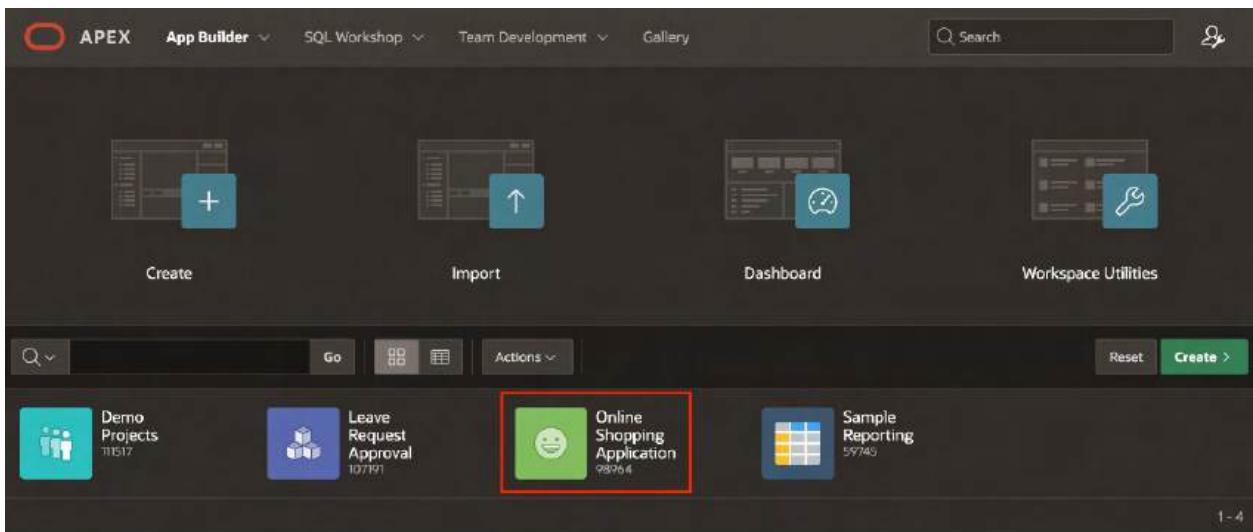
24. You are now logged in to the **Online Shopping Application**. Check the user name on the top right of the **Navigation Bar**.



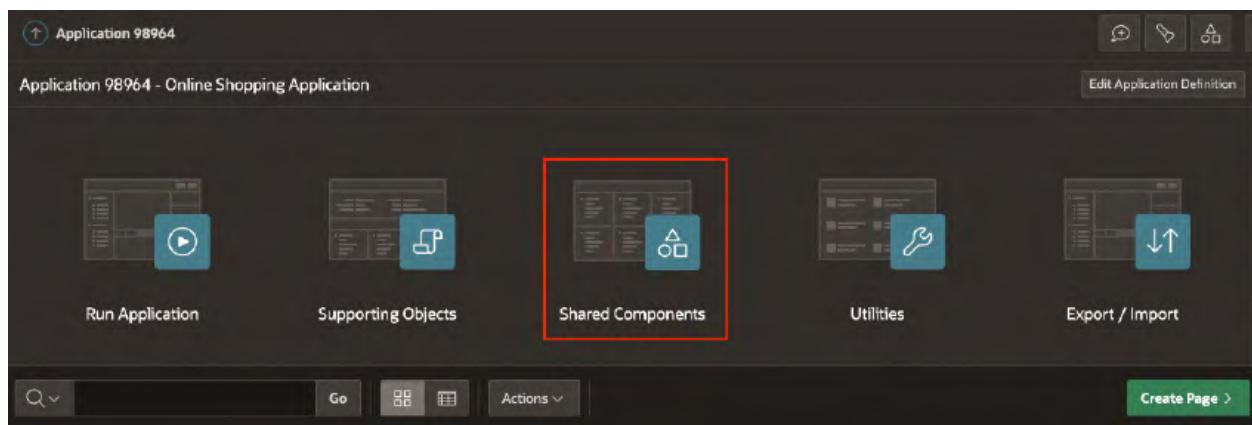
Creating and Using an Authorization Scheme

In this task, you create an **Authorization Scheme** to ensure only people entered as Team Members can log into the **Online Shopping Application**. You apply the authorization scheme to the application properties.

1. Navigate to App Builder and select **Online Shopping Application**.



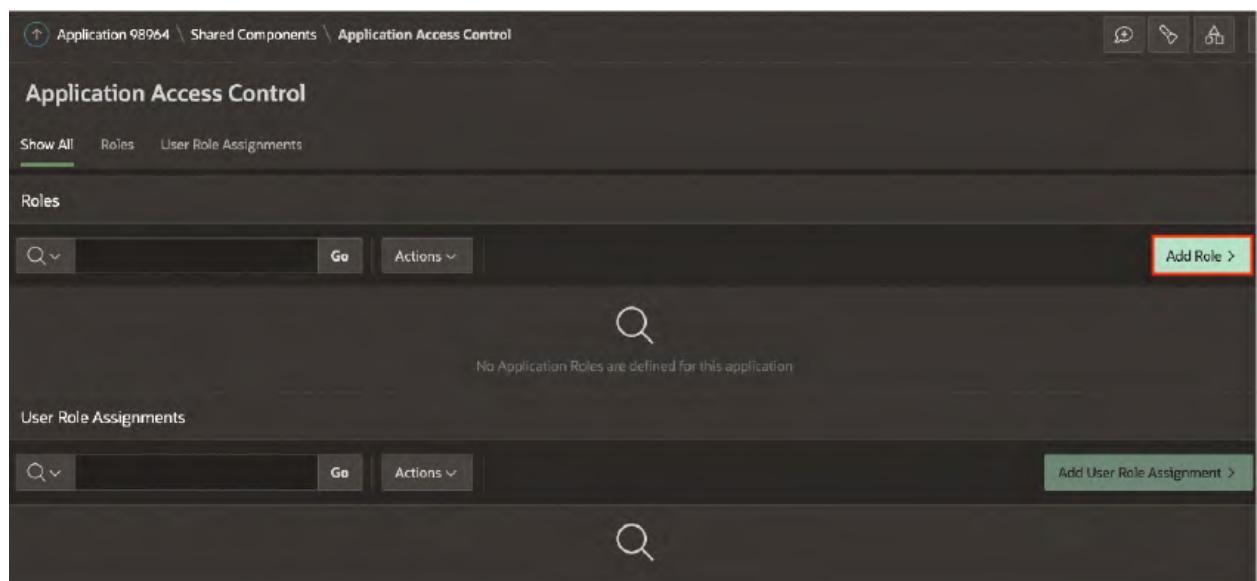
2. In the application home page, click **Shared Components**.



3. Under Shared Components > Security, click **Application Access Control**.



4. Under **Roles**, Select **Add Role**.



5. In the **Role** Page, enter the following and click **Create Role**.

- a. For **Name**, Enter **Administrator**
- b. For **Static Identifier**, select **ADMINISTRATOR**

The screenshot shows the 'Role' creation dialog box. At the top, it says 'Role'. Below that, there are two required fields: 'Name' (containing 'Administrator') and 'Static Identifier' (containing 'ADMINISTRATOR'). There is also a 'Description' field which is empty. Underneath these fields, there is a section titled 'Associated Authorization Schemes' with a warning message: '⚠️ Changing the Name or Static Identifier of this Role may cause the following associated authorization schemes to fail.' At the bottom left is a 'Cancel' button, and at the bottom right is a green 'Create Role' button, which is highlighted with a red border.

6. In the **User Role Assignments**, Click **Add User Role Assignment**.

The screenshot shows the Oracle Application Access Control interface. At the top, there's a navigation bar with 'Application 98964 \ Shared Components \ Application Access Control'. Below it, a green header bar says 'Action Processed' with a checkmark icon. The main area has two tabs: 'Show All' (selected), 'Roles', and 'User Role Assignments'. Under 'Show All', there's a 'Roles' section with a table:

| Role ↑ | Static Identifier | Users | Description |
|---------------|-------------------|-------|-------------|
| Administrator | ADMINISTRATOR | 0 | - |

Below this is a 'User Role Assignments' section with a table:

| Role ↑ | Actions |
|--------|---|
| | Add User Role Assignment > |

A red box highlights the 'Add User Role Assignment >' button.

7. In the **User Assignment** Page, enter the following and click **Create Assignment**.

- For **User Name**, Enter **AUTHORIZED USER**.
- For **Application Role**, Check **ADMINISTRATOR** to Yes

The screenshot shows the 'User Assignment' dialog box. It has a warning message at the top:

About Application Users
Application users are not exported as part of your application. When you deploy your application you will need to manually manage your user to role assignments. Roles are exported as part of an application export and imported with application imports.

Below the message, there are fields to enter user information:

* User Name: AUTHORIZED USER

Application Role: Administrator

At the bottom, there are 'Cancel' and 'Create Assignment' buttons. A red box highlights the 'Create Assignment' button.

8. Navigate back to **Shared Components** and then select **Authorization Schemes**.

The screenshot shows the 'Application Access Control' page. At the top, there are tabs for 'Shared Components' (which is selected) and 'Application Access Control'. Below the tabs, there are sections for 'Roles' and 'User Role Assignments'. The 'Roles' section lists an 'Administrator' role with a static identifier 'ADMINISTRATOR'. The 'User Role Assignments' section shows an assignment for 'AUTHORIZED USER' to the 'Administrator' role.

| Role | Static Identifier | Users | Description |
|---------------|-------------------|-------|-------------|
| Administrator | ADMINISTRATOR | 1 | |

| User Name | Roles |
|-----------------|---------------|
| AUTHORIZED USER | Administrator |

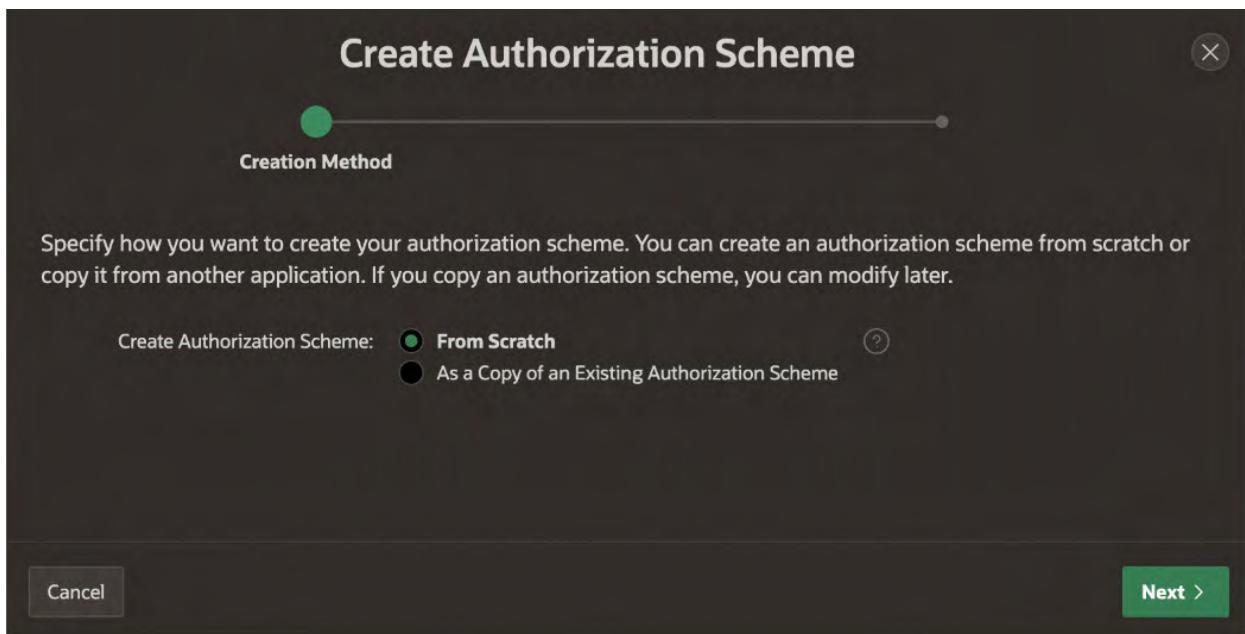
The screenshot shows the 'Shared Components' navigation menu. It includes sections for Application Logic (Application Definition, Application Items, Application Processes, Application Computations, Application Settings, Build Options), Security (Security Attributes, Authentication Schemes, Authorization Schemes, Application Access Control, Session State Protection), and Other Components (Lists of Values, Plug-ins, Component Settings, Shortcuts). The 'Authorization Schemes' link under Security is highlighted with a red box.

9. Under **Authorization Scheme**, Click **Create**.

The screenshot shows the 'Authorization Schemes' creation page. The 'Authorization Schemes' tab is selected. The page has a search bar, a toolbar with 'Copy', 'Reset', and 'Create >' buttons, and a table for listing schemes. A new scheme named 'Administration Rights' is listed, with a type of 'PL/SQL Function Returning Boolean' and a caching policy of 'Once per page view'. The 'Create >' button is highlighted with a green box.

| Name | Type | Caching | Subscribed From | Subscribers | Updated |
|-----------------------|-----------------------------------|--------------------|-----------------|-------------|---------|
| Administration Rights | PL/SQL Function Returning Boolean | Once per page view | | | |

10. For **Create Authorization Scheme**, Leave the settings to default and then click **Next**.



11. In the **Details** section of Create Authorization Scheme, enter the following and click **Create Authorization Scheme**.

- a. For **Name**, Enter **Admin**.
- b. For **Scheme Type**, Select **Is In Role or Group**.
- c. For **Type**, select **Application Role**
- d. For **Name(s)**, Select **Administrator**.
- e. For **Identify error message displayed**, enter **You are not Authorized to view this**.

Create Authorization Scheme

Details

Use this page to define an authorization scheme. By creating an authorization schemes, you can protect applications, pages, and application components and extend the security provided by your application authentication scheme. You can use authorization schemes to identify additional security beyond simple user authentication. For example a user with administration rights may need access to more navigation bar icons, pages, and tabs than other users.

Application: **98964 Online Shopping Application**

Name: **Admin**

Scheme Type: **Is In Role or Group**

Type: **Application Role**

Name(s): **Administrator**

Identify error message displayed when scheme violated: **You are not Authorised to view this.**

Validate authorization scheme: **Once per session**

Comments:

Create Authorization Scheme

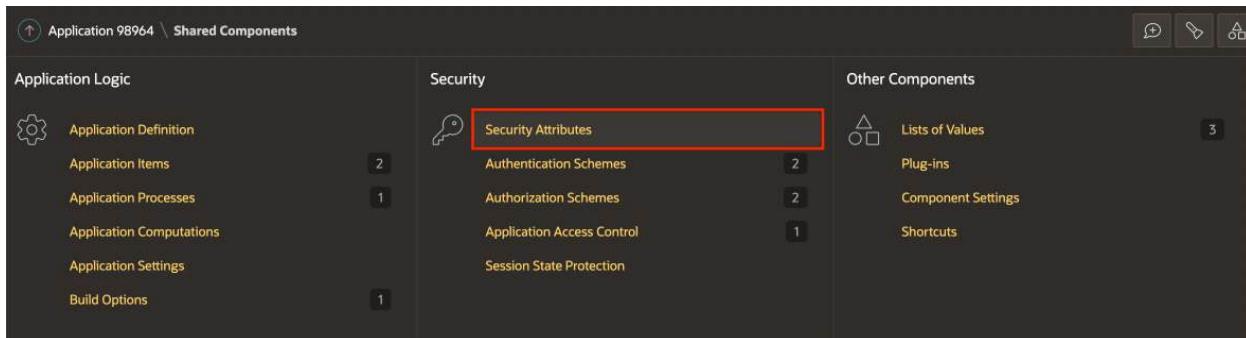
12. Navigate to **Shared Components** and then under **Security**, Select **Security Attributes**.

Application 98964 **Shared Components** Authorization Schemes

Authorization Schemes Subscription by Component Utilization History

Actions: Copy Reset Create >

| Name | Type | Caching | Subscribed From | Subscribers | Updated |
|-----------------------|-----------------------------------|--------------------|-----------------|-------------|---------------|
| Admin | Is In Role or Group | Once per session | | | 1 seconds ago |
| Administration Rights | PL/SQL Function Returning Boolean | Once per page view | | | |

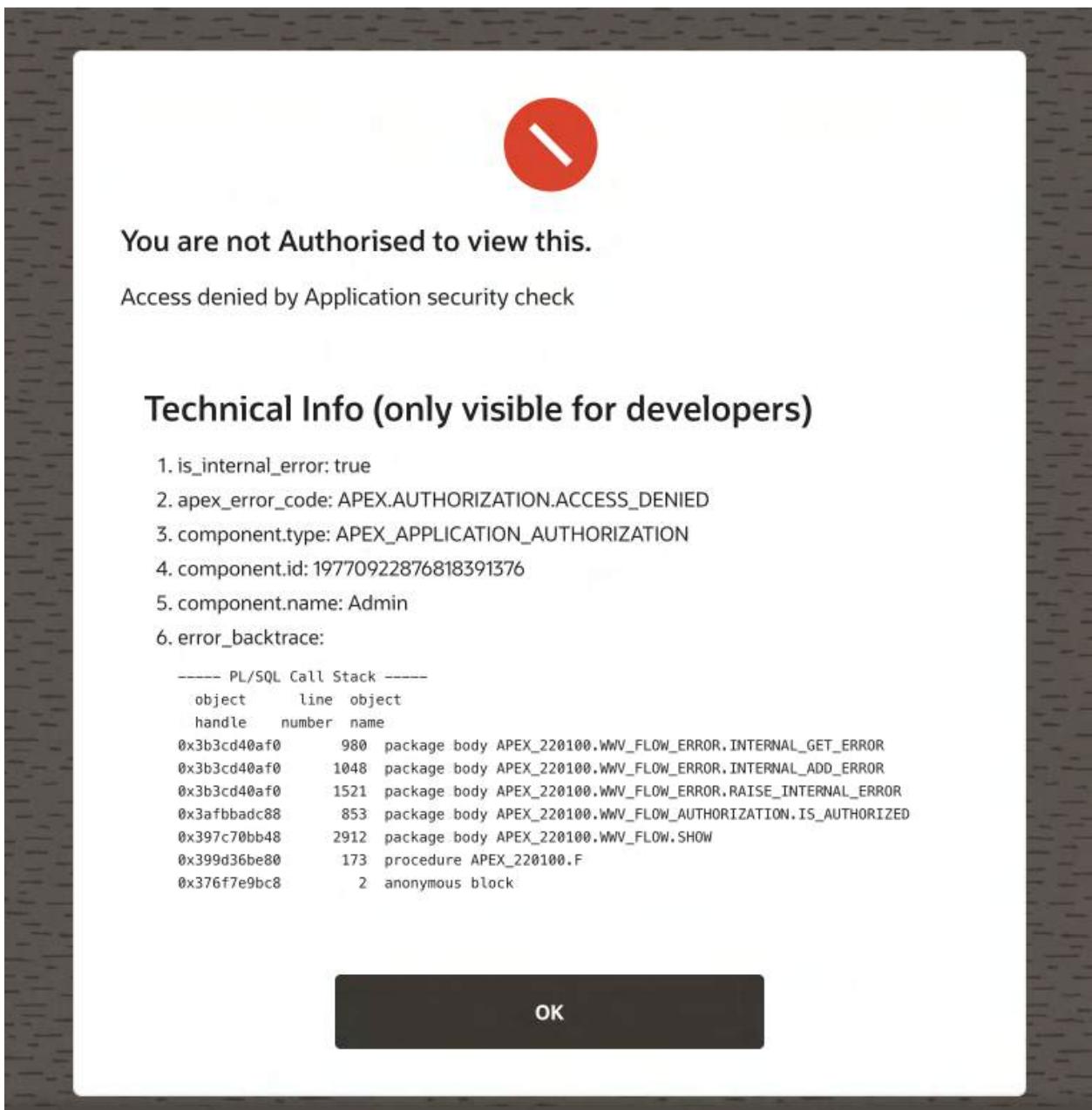


13. In the **Edit Security Attributes** Page, under **Authorization**, select **Admin** for **Authorization Scheme**. Click **Apply Changes**.

| Setting | Value |
|-----------------------|----------|
| Authorization Scheme | Admin |
| Run on Public Pages | Disabled |
| Run on Background Job | Enabled |

14. Navigate to the runtime environment (tab or window). Click Sign Out in the navigation bar (top right).
15. Run the application by navigating to **Online Shopping Application** and click Run. In the User Interface, click **Administration**.

16. Since your name is not **Authorized User**, verify the access denied message is displayed.



Summary

You now know how to create a Social Sign-in authentication scheme to enable Facebook Authentication

Practice: Adding Additional Pages to your Application

Practice 1: Adding Additional Pages to Your Application

Overview

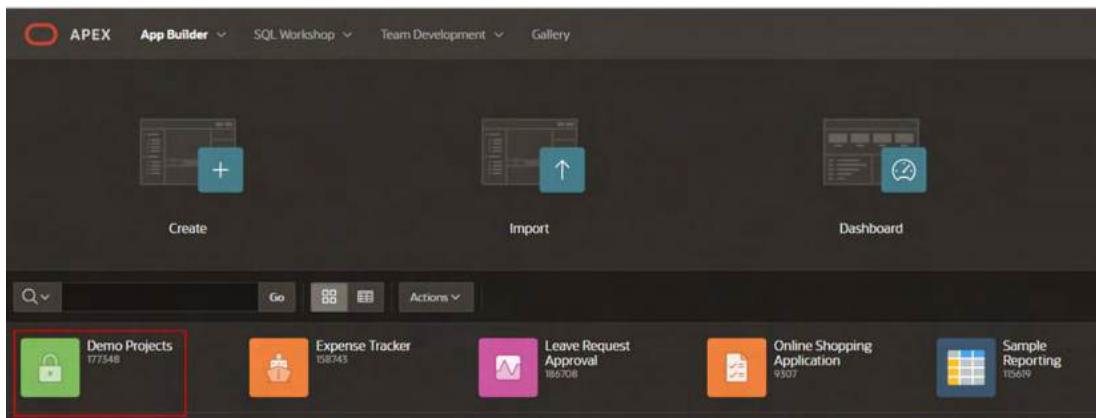
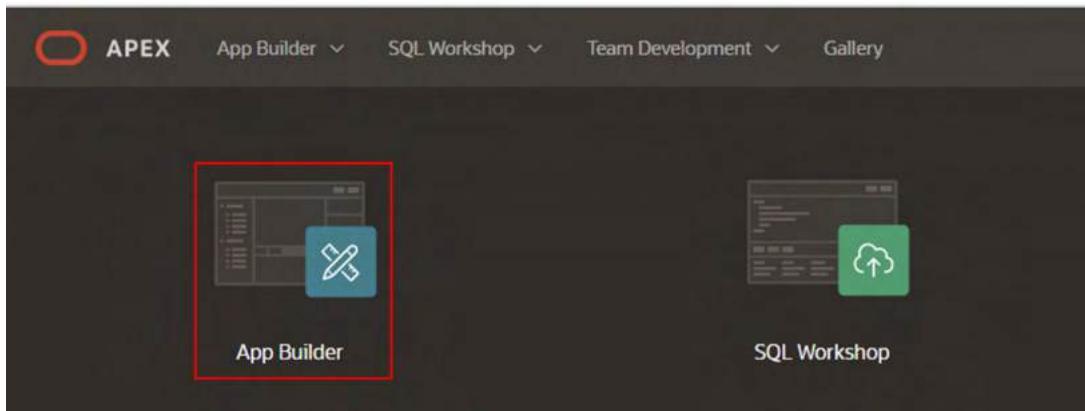
In this hands-on lab, you will add Calendars, Charts, Tree pages, and Maps to the Demo Projects application and to the Online Shopping application.

Downloads

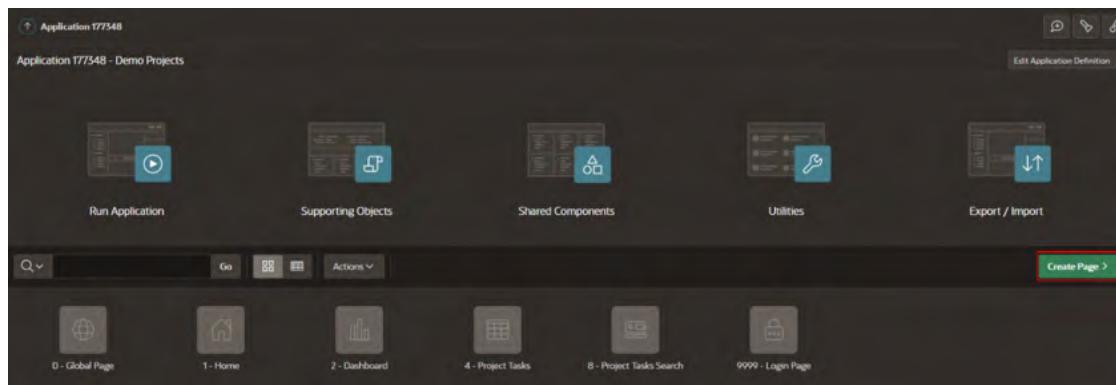
- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Create a Calendar

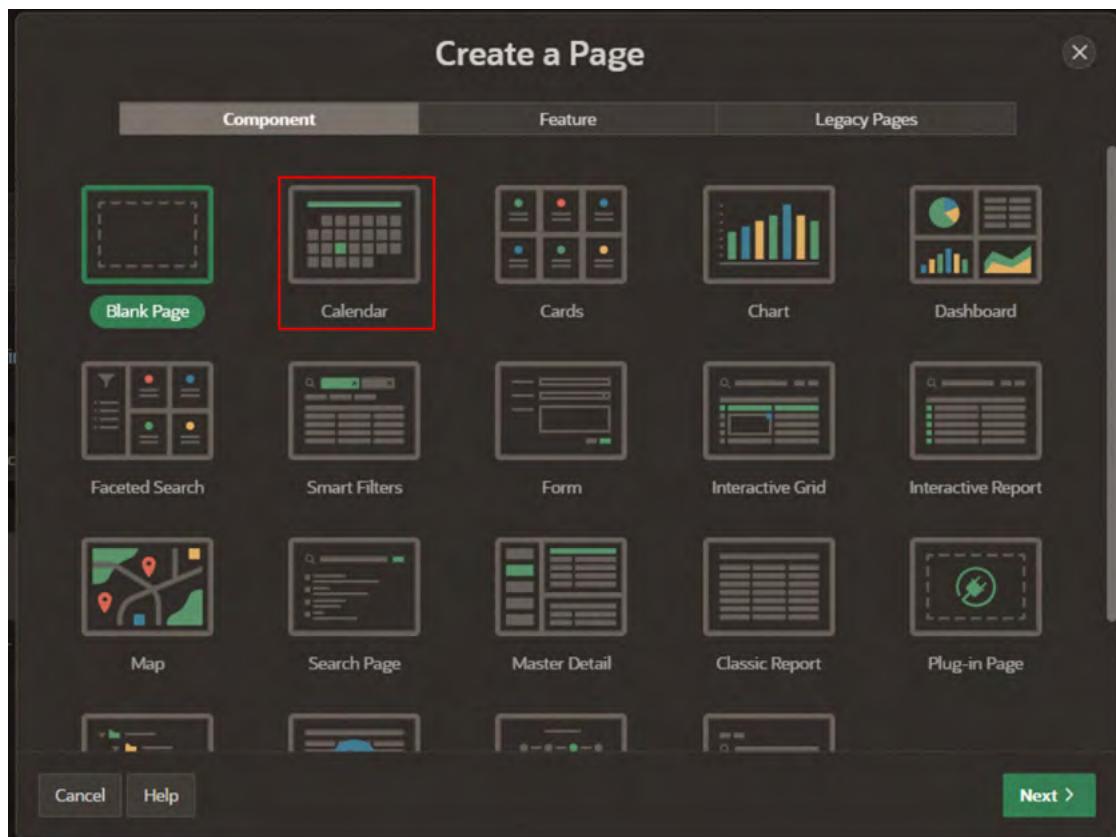
1. Navigate to **App Builder** and in the **Home Page**, click **Demo Projects**.



2. In the application home page, click **Create Page**.



3. Select **Calendar** page type.



4. In the **Create Calendar** page, enter the following and click **Next**:

Under Page Definition:

- For **Name**, Enter **Calendar**.

Under Data Source:

- For Table / View Name, select DEMO_PROJECTS.

Under Navigation:

- Set Breadcrumb and Navigation to Yes.

Create Calendar

Page Definition

* Page Number [?](#)

* Name [?](#)

Page Mode [Normal](#) [Modal Dialog](#) [Drawer](#) [?](#)

Data Source

Data Source [Local Database](#) [REST Enabled SQL Service](#) [REST Data Source](#) [?](#)

Source Type [Table](#) [SQL Query](#) [?](#)

* Table / View Owner [?](#)

* Table / View Name [?](#)

Navigation

Use Breadcrumb [?](#)

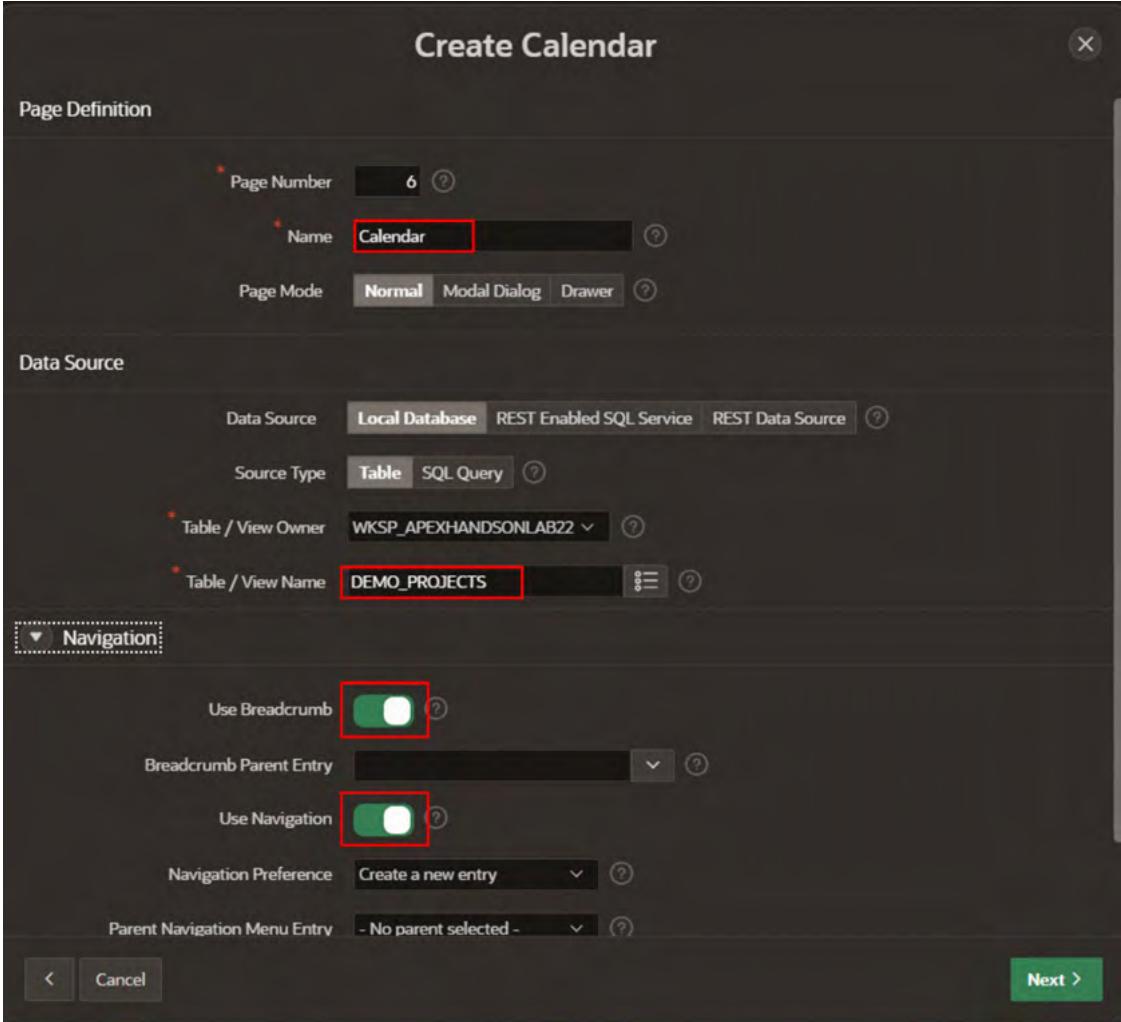
Breadcrumb Parent Entry [?](#)

Use Navigation [?](#)

Navigation Preference [?](#)

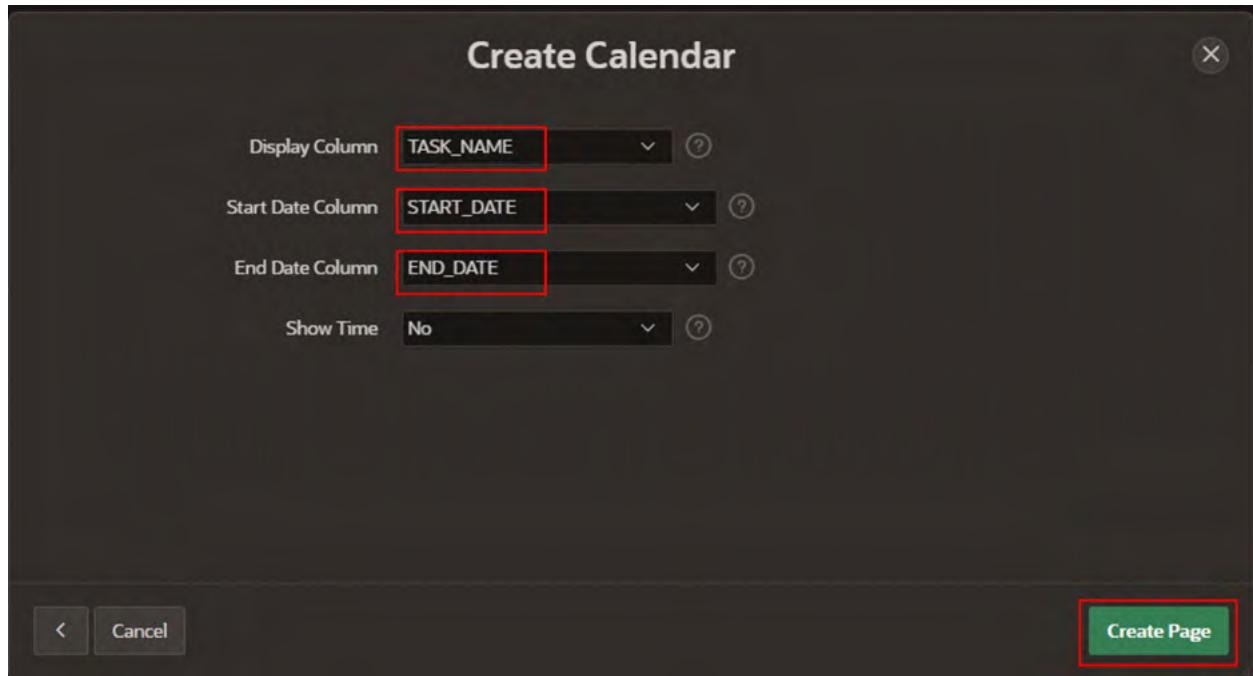
Parent Navigation Menu Entry [?](#)

[<](#) [Cancel](#) [Next >](#)



5. In the **Create Calendar** page, enter the following and click **Create Page**.

- For **Display Column**, select **TASK_NAME**.
- For **Start Date Column**, select **START_DATE**.
- For **End Date Column**, select **END_DATE**.

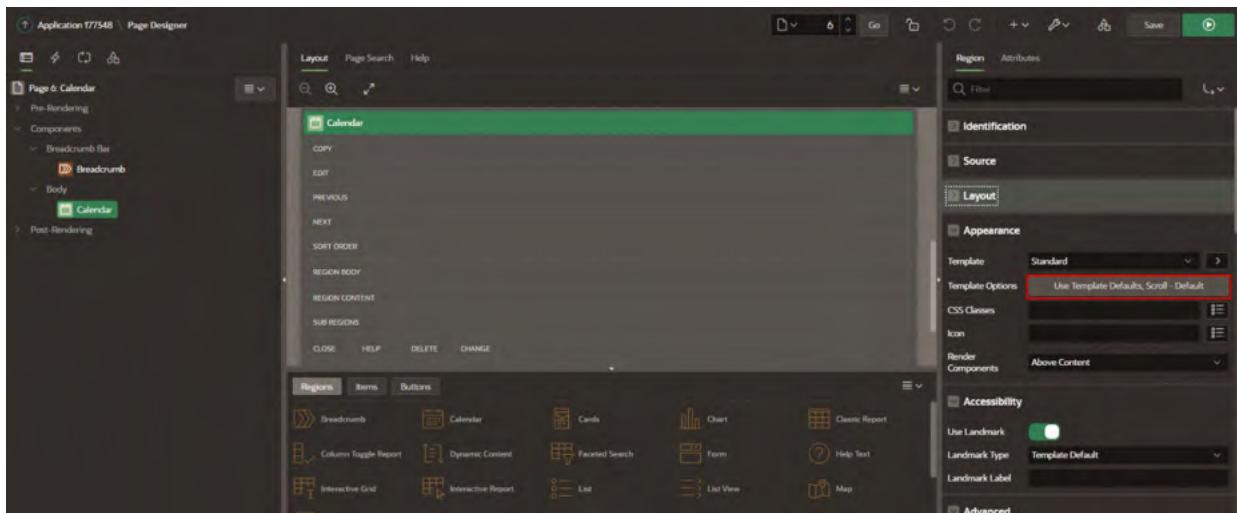


6. Click **Save** and **Run Page**. Log in to the application with your credentials.

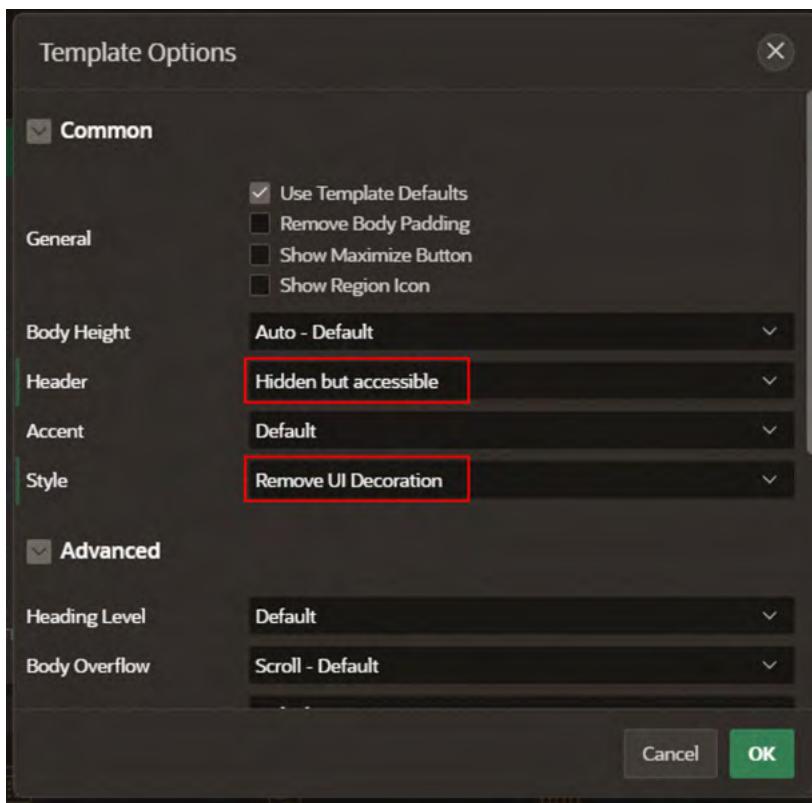
7. In the Developer Toolbar, click **Edit Page**.

The screenshot shows a calendar component for March 2023. The days of the week are labeled from Sunday to Saturday. The dates range from 1 to 25. The days 5 and 6 are highlighted with a yellow background, indicating they are selected or part of a specific event. The developer toolbar at the bottom of the page is visible, showing various icons and options. The 'Edit Page' icon is highlighted with a red border.

- The Calendar page displays the **region title Calendar**, and also has a border around the region. In the Rendering tree, locate the Calendar region. Click **Calendar**. In the **Property Editor**, under **Appearance**, click the **Template Options** button.



- In the Template Options dialog box, input the following:
 - Header - select **Hidden but accessible**.
 - Style - select **Remove UI Decoration**.



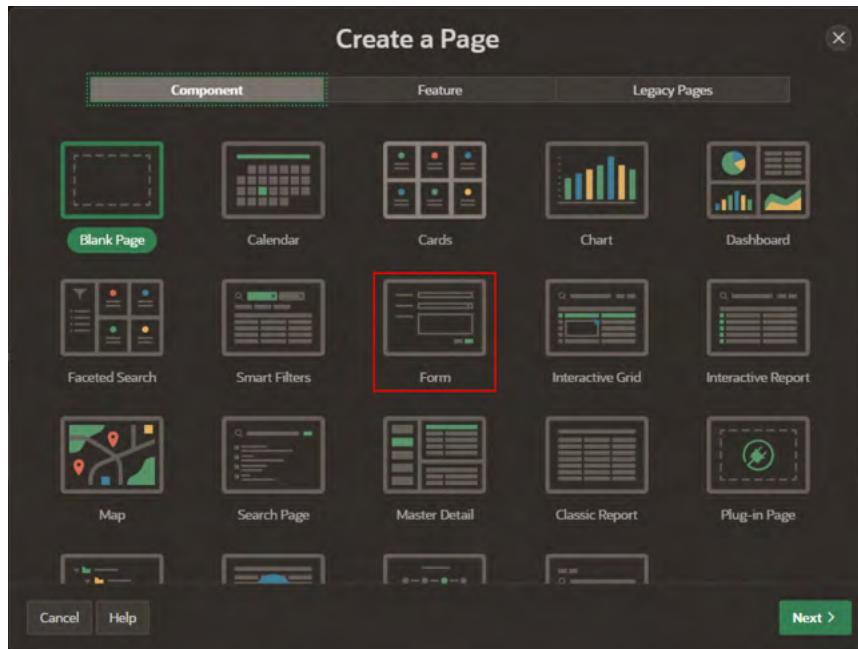
Create a Form Page on DEMO_PROJECTS Tables

In this lab, you will create a Form Page on the DEMO_PROJECTS tables, then in the next labs, you will link the form to the Calendar Page.

1. Navigate to **App Builder** and in the **Home Page**, click **Demo Projects**. Then, click **Create Page**.



2. Select the **Form** page type.



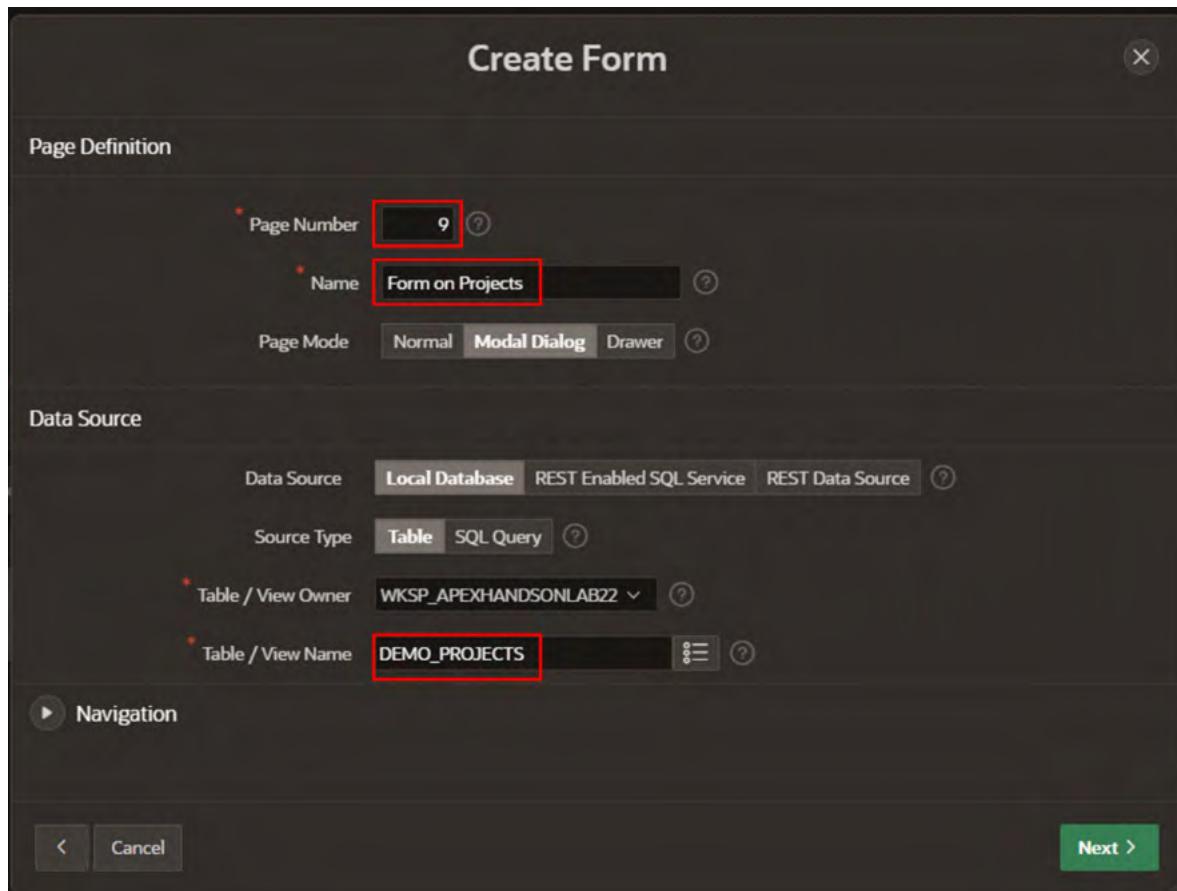
3. In the **Create Form**, enter the following and click **Next**.

Under Page Definition:

- For **Page Number**, enter 9.
- For **Name**, enter **Form on Projects**.
- For Page Mode, select **Modal Dialog**.

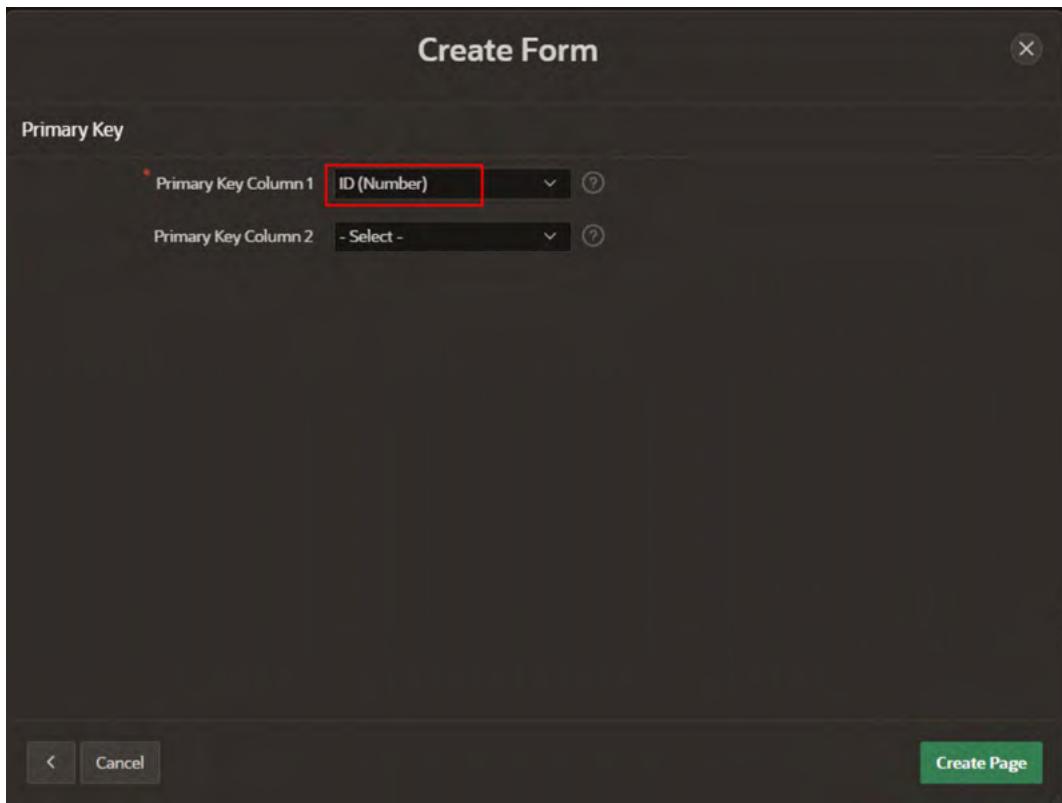
Under Data Source:

- For **Table / View Name**, select **DEMO_PROJECTS**.



4. In the **Create Form** page, enter the following and click **Create Page**.

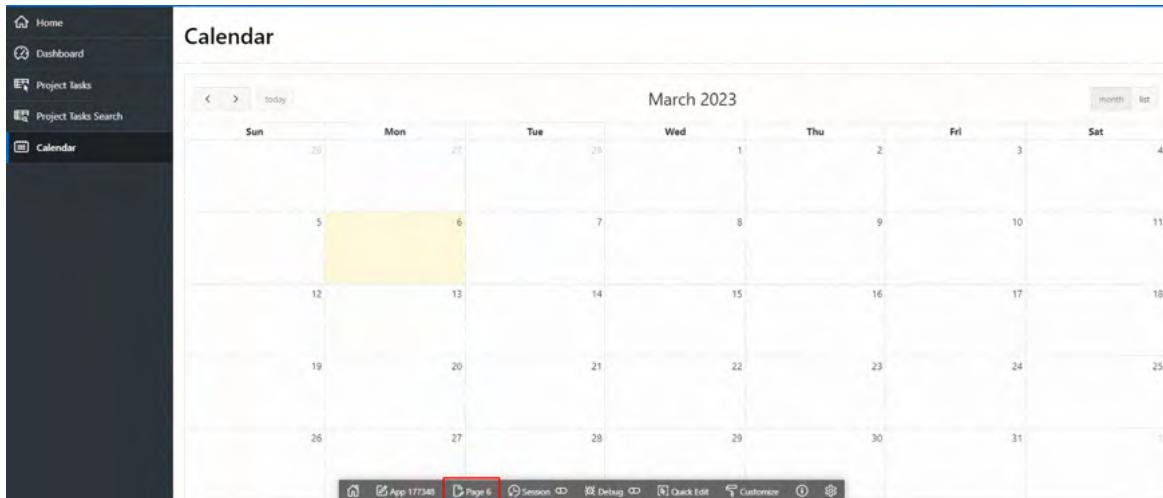
- For **Primary Key Column 1**, select **ID (Number)**.



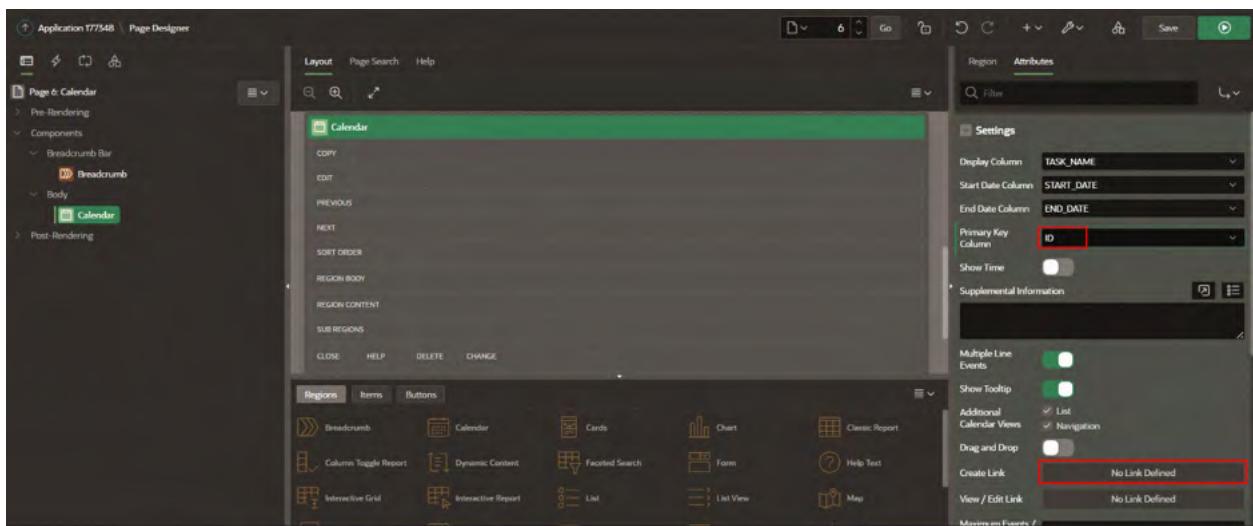
Customize the Calendar Page

In this lab, you will link the form page you created in Task 2 with the Calendar page.

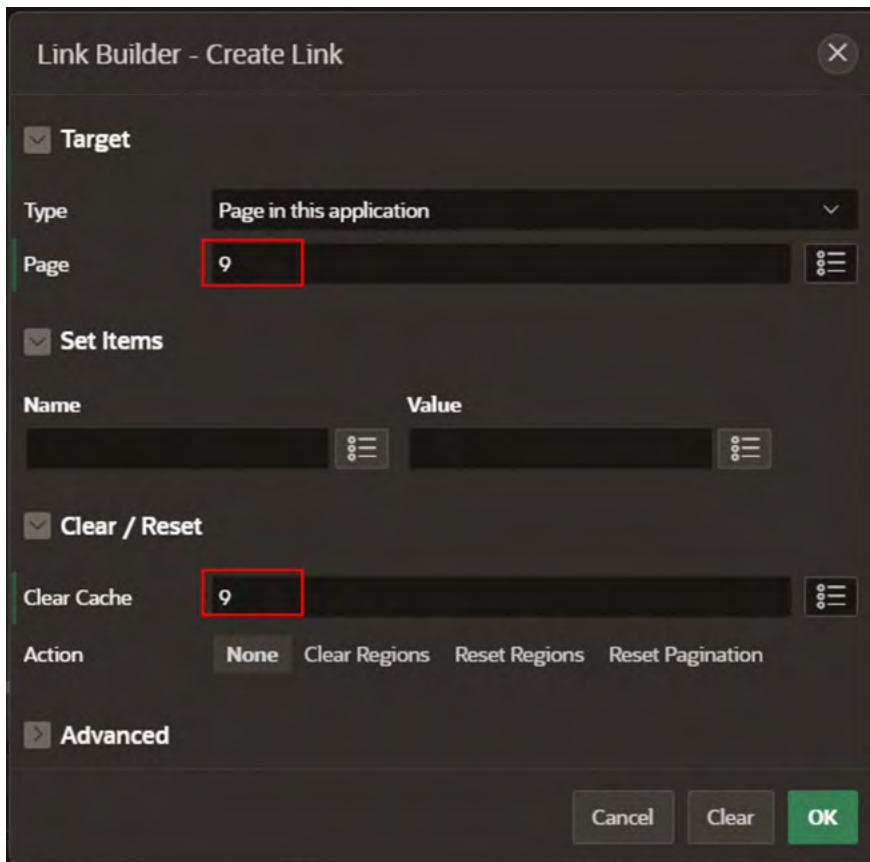
1. Navigate to **Calendar** in the runtime environment and then click **Page**.



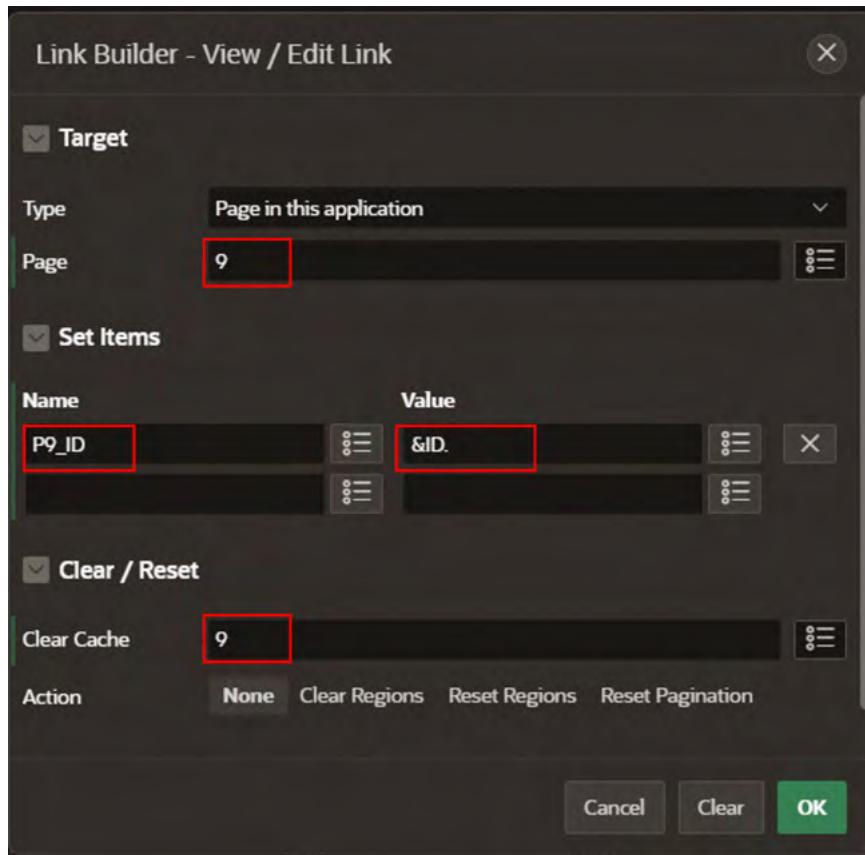
2. You need to add the **Create** and **View / Edit** links. In the Rendering tree, locate and select the **Calendar** region. In the **Property Editor**, click **Attributes**. Then, Under Settings, Select **ID** for Primary Key and then locate **Create Link** and click **No Link Defined**.



3. In the Link Builder – Create Link dialog box, select **9** for Page, and enter **9** for **Clear Cache**. Click **OK**.



- In the Property Editor, locate **View/Edit Link** and click **No Link Defined**.
- In the Link Builder – View / Edit Link dialog box, input the following:
 - Page - select **9**.
 - Name - select **P9_ID**
 - Value - select **ID** or enter **&ID.**.
 - Clear Cache - enter **9**.

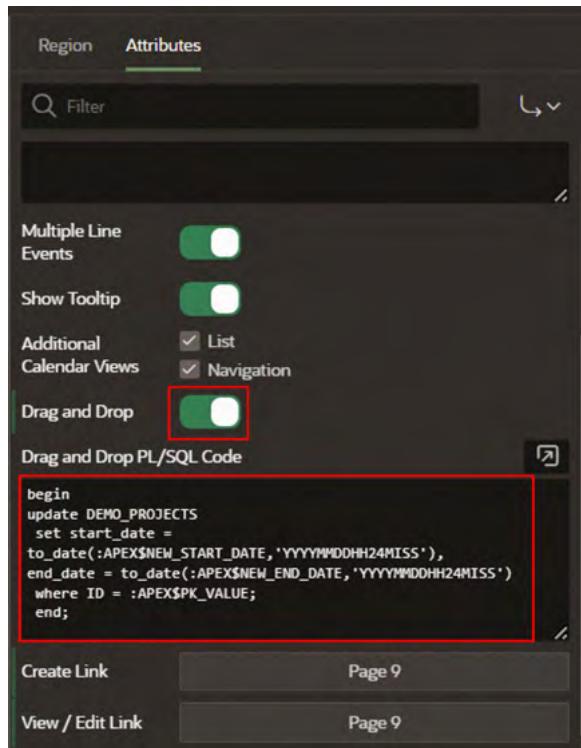


- You can enable calendar drag and drop by using the component attribute **Drag and Drop**. Your SQL query must select a primary key column, and you must have set the Primary Key Column calendar attribute. Then enter the PL/SQL code to update the event row in the database in the Drag and Drop PL/SQL Code attribute. That PL/SQL code typically performs a SQL update on the database table - the bind variables:
:APEX\$PK_VALUE, **:APEX\$NEW_START_DATE**, and **:APEX\$NEW_END_DATE** contain the dragged events primary key value as well as the new start and new end timestamp.

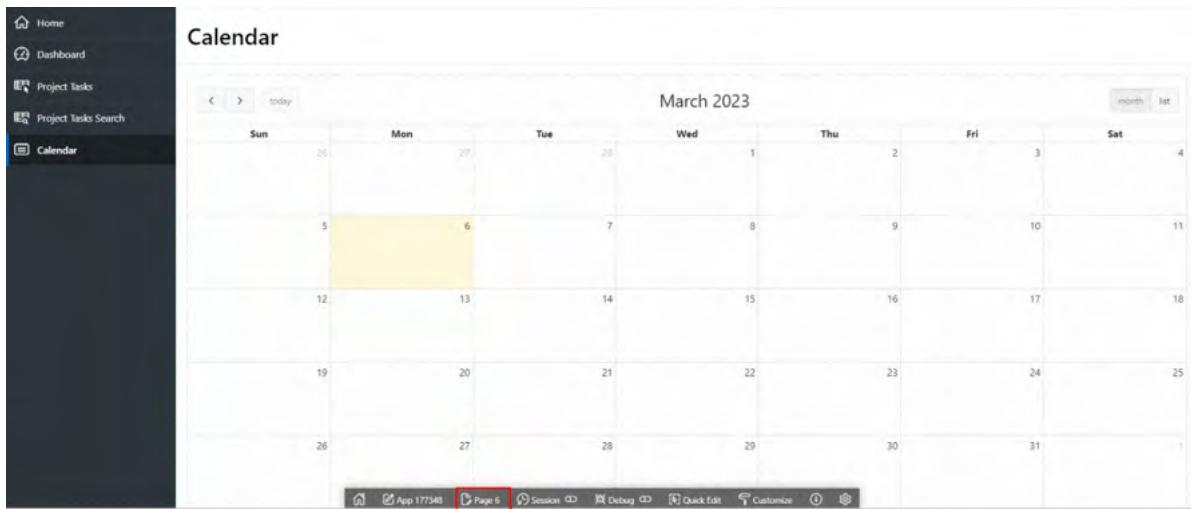
Under **Settings**:

- For **Drag and Drop**, set it to **Yes**.
- For **Drag and Drop PL/SQL Code**, copy and paste the below code:

```
begin
  update DEMO_PROJECTS
  set start_date =
  to_date(:APEX$NEW_START_DATE, 'YYYYMMDDHH24MISS'),
  end_date = to_date(:APEX$NEW_END_DATE, 'YYYYMMDDHH24MISS')
  where ID = :APEX$PK_VALUE;
end;
```



- Click **Save** and **Run Page**. Notice that you can now drag and drop tasks in the calendar. In the Developer Toolbar, click **Application< n >**



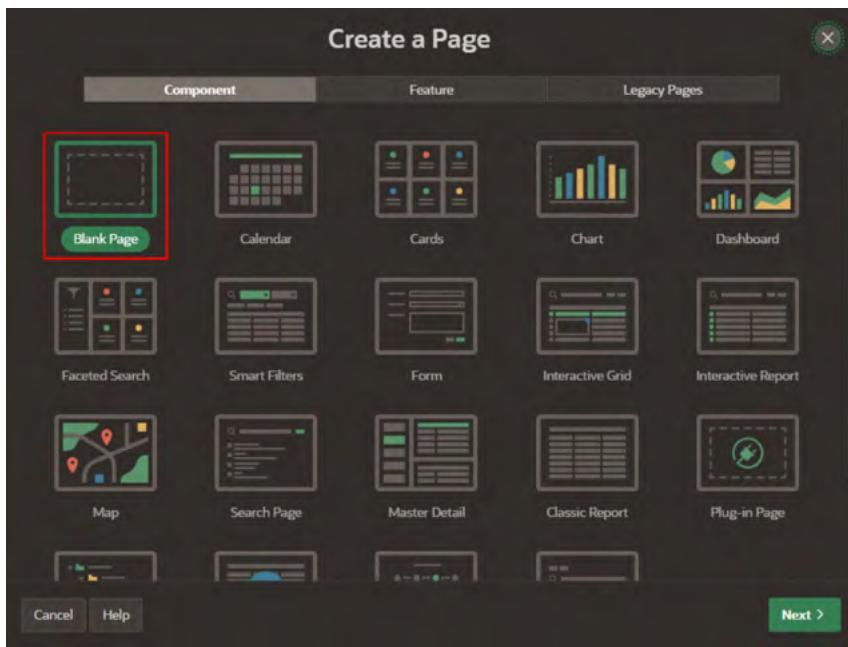
Create and Customize a Tree Page

In this hands-on lab, you create the **Employee** tree by first creating a **blank page** and then adding a **Tree region**.

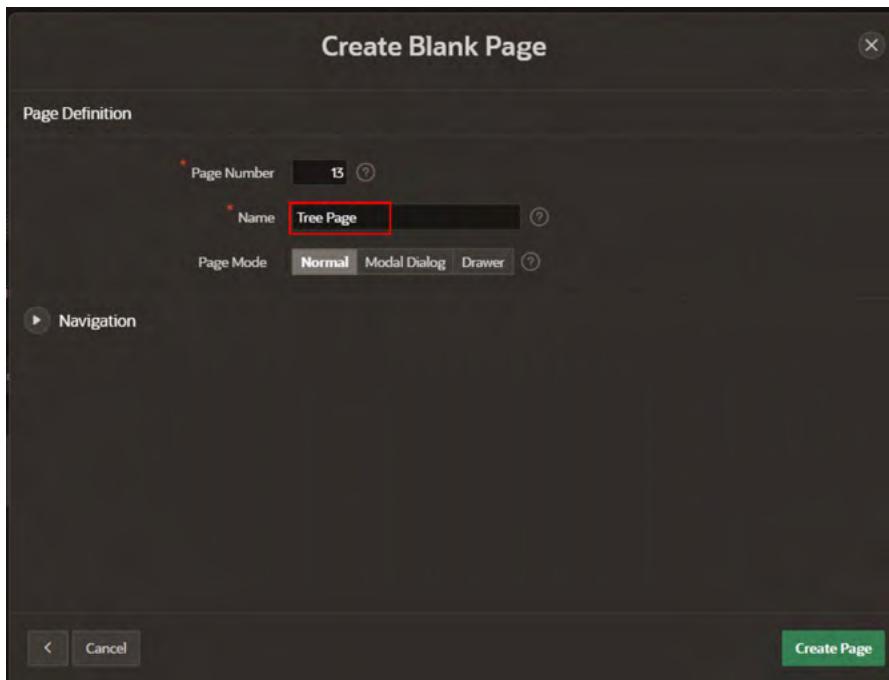
- First, create a blank page in the **Demo Projects** application. In the application home page, click **Create Page**.



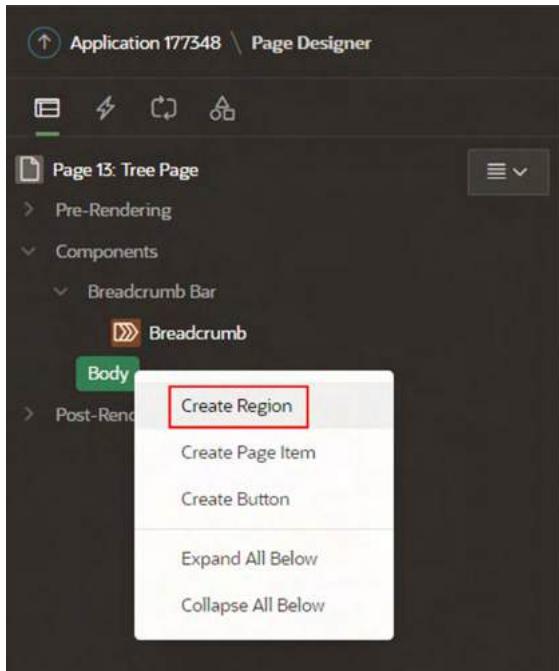
2. Select **Blank Page**.



3. Enter **Tree Page** for **Name** and then click **Create Page**.



4. Now, you create a **Tree region**. In the page designer, under Rendering, right-click **Body** and select **Create Region**.



5. In the property editor, enter the following:

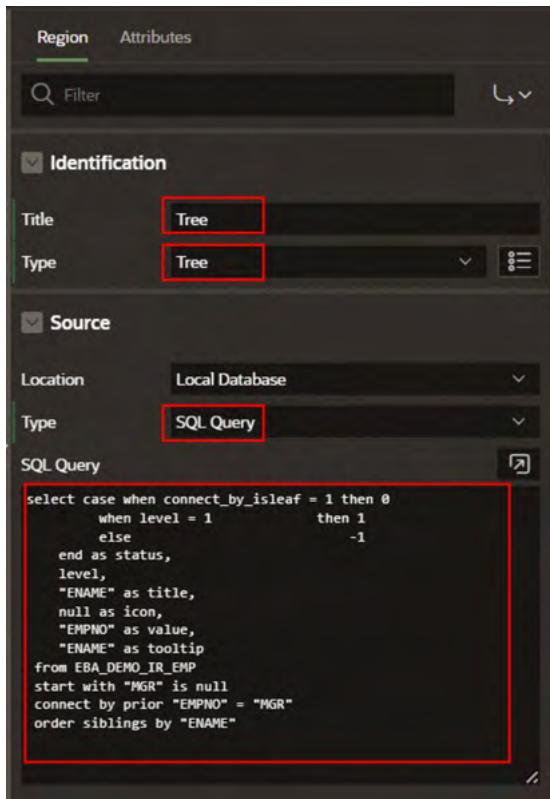
Under Identification:

- For **Name**, enter **Tree**.
- For **Type**, select **Tree**.

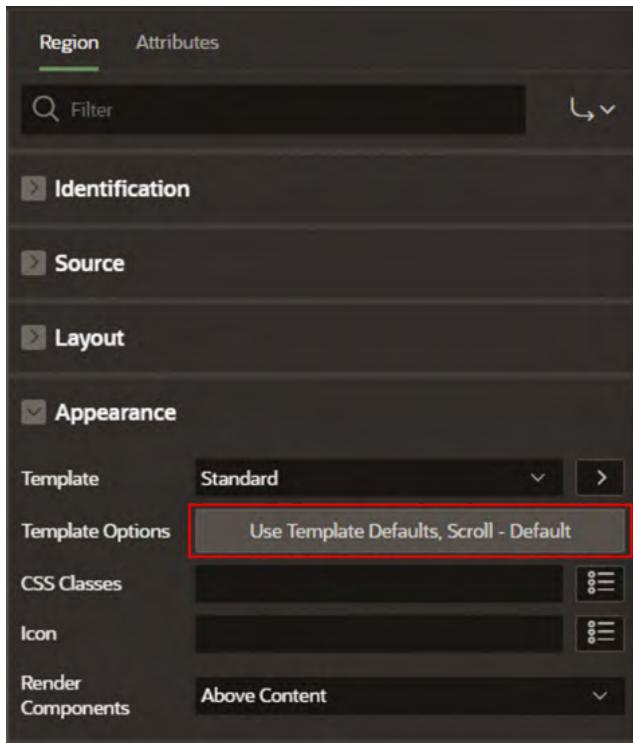
Under **Source**:

- For **Type**, select **SQL Query**.
- For **SQL Query**, copy the following code and paste it:

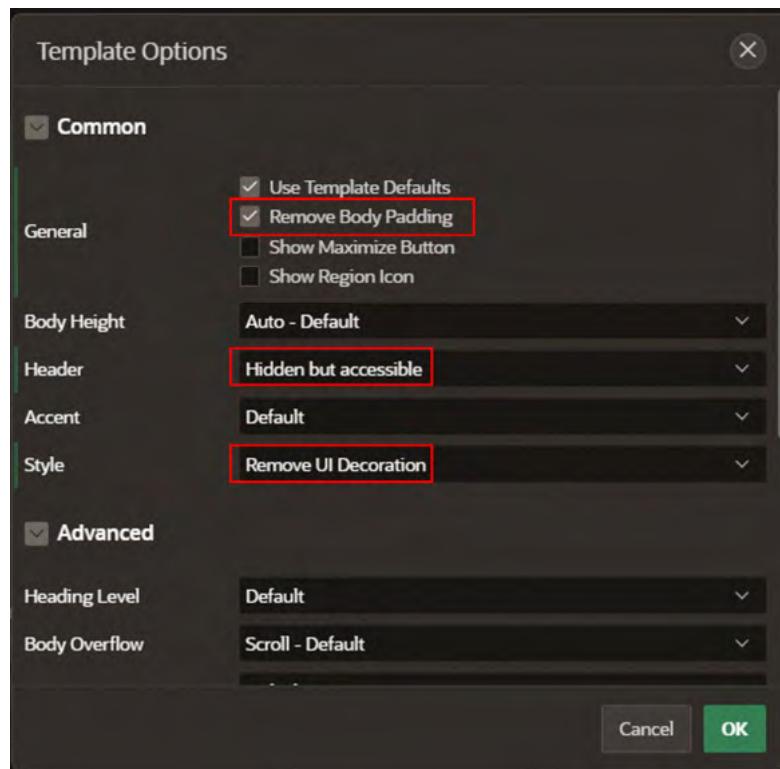
```
select case when connect_by_isleaf = 1 then 0
           when level = 1                  then 1
           else                            -1
      end as status,
      level,
      "ENAME" as title,
      null as icon,
      "EMPNO" as value,
      "ENAME" as tooltip
   from EBA_DEMO_IR_EMP
  start with "MGR" is null
 connect by prior "EMPNO" = "MGR"
 order siblings by "ENAME"
```



- In the page designer, navigate to **Appearance** and then click the **Template Options** button.

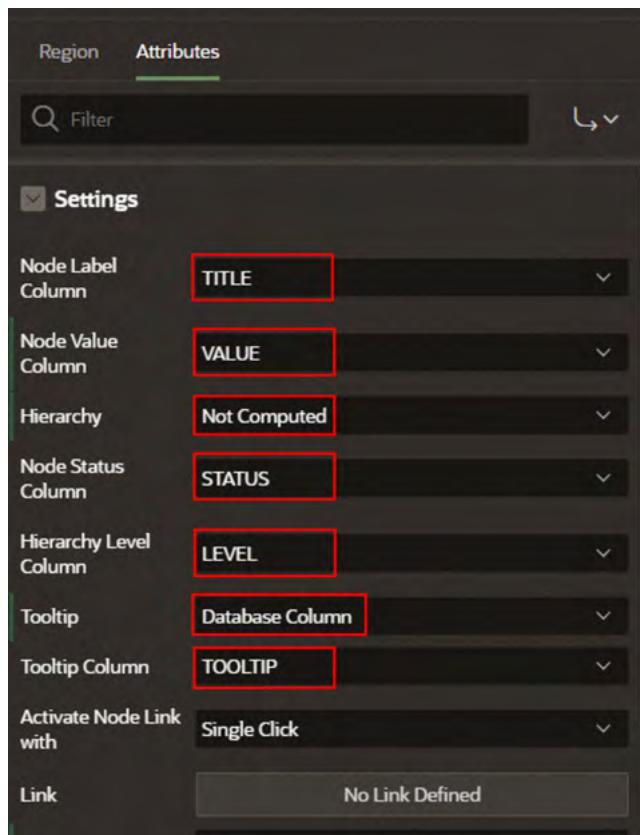


7. In the Template Options dialog box:
- General: Select the **Remove Body Padding** check box.
 - Header: Select **Hidden but accessible**.
 - Style: Select **Remove UI Decoration**. Click **OK**.



8. In the **Property Editor**, Select **Attributes**. Navigate to **Settings** and select / enter the following:
- Node Label Column: **TITLE**
 - Node Value Column: **VALUE**
 - Hierarchy: **Not Computed**
 - Node Status Column: **STATUS**
 - Hierarchy Level Column: **LEVEL**
 - Tooltip: **Database Column**
 - Tooltip Column: **TOOLTIP**

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



9. The **Tree Page** is now displayed.

Tree Page



You now know how to add Calendars and Tree pages.

You may now **proceed to the next lab**.

Practice 2: Add Map to Your Application

Overview

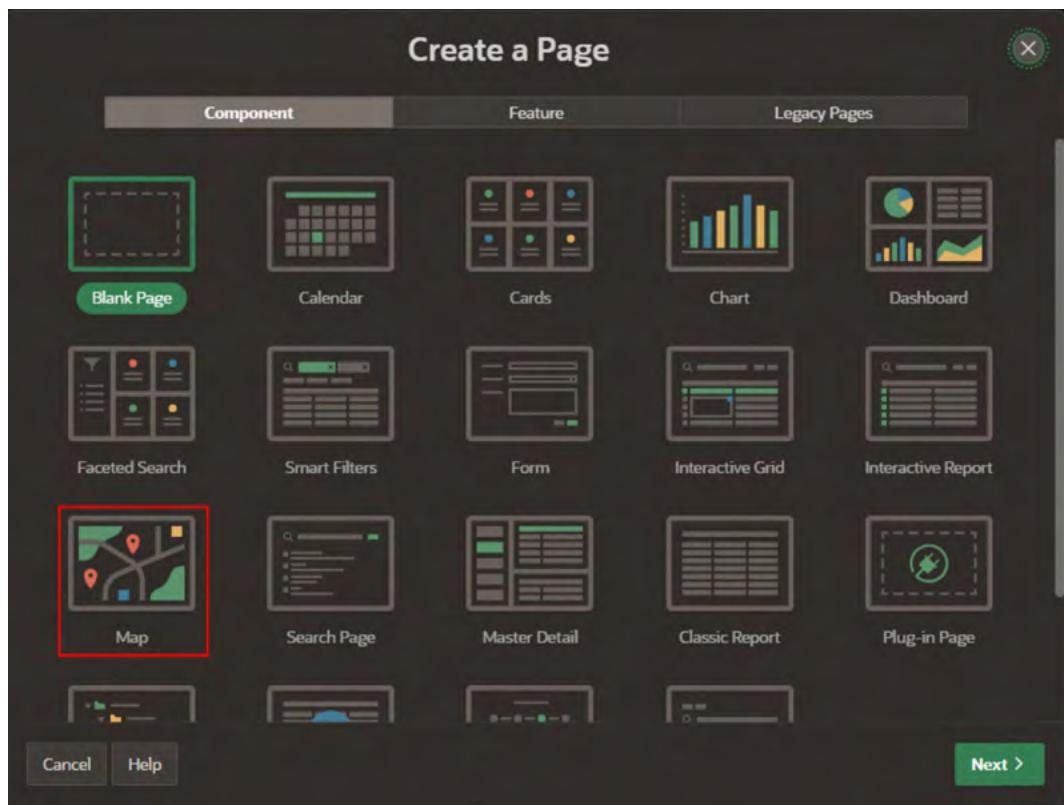
In this lab, you will first create a Map Page with Store Details, and then you will create an entry for the Store Details Map in the navigation Menu Entry.

Downloads

- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Create a Store Details Map Page and Add It to Desktop Navigation Bar

- Navigate to **App Builder** and in the **Home Page**, click **Online Shopping Application**.
- In the application home page, click **Create Page**.
- Select **Map** page type.



4. In the **Create Map** page, enter the following and click **Next**.

Under Page Definition:

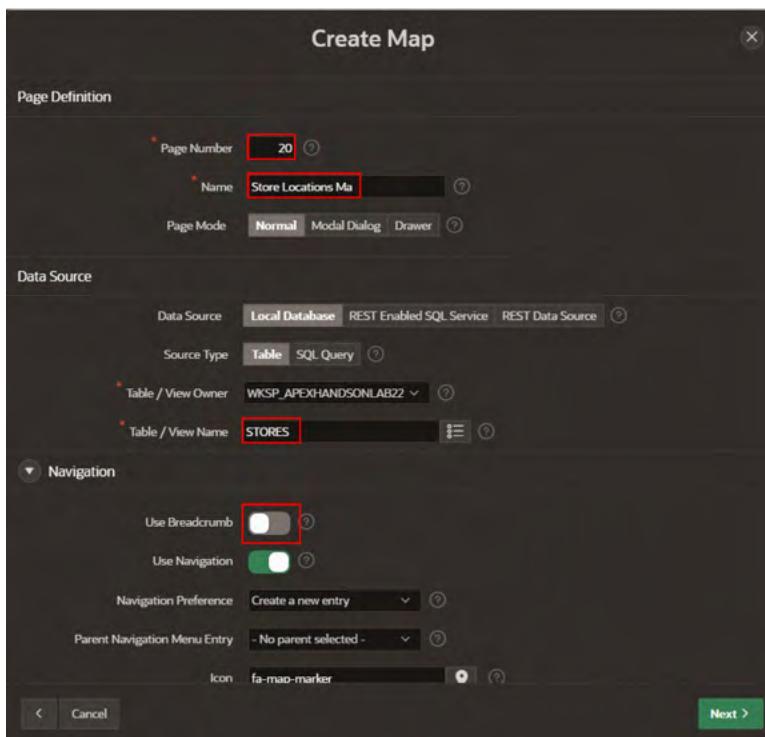
- For **Page Number**, enter 20.
- For **Name**, enter **Store Locations Map**.

Under Data Source:

- For **Table/View Name**, select **STORES**.

Under Navigation:

- For **Breadcrumb**, set it to **No**.

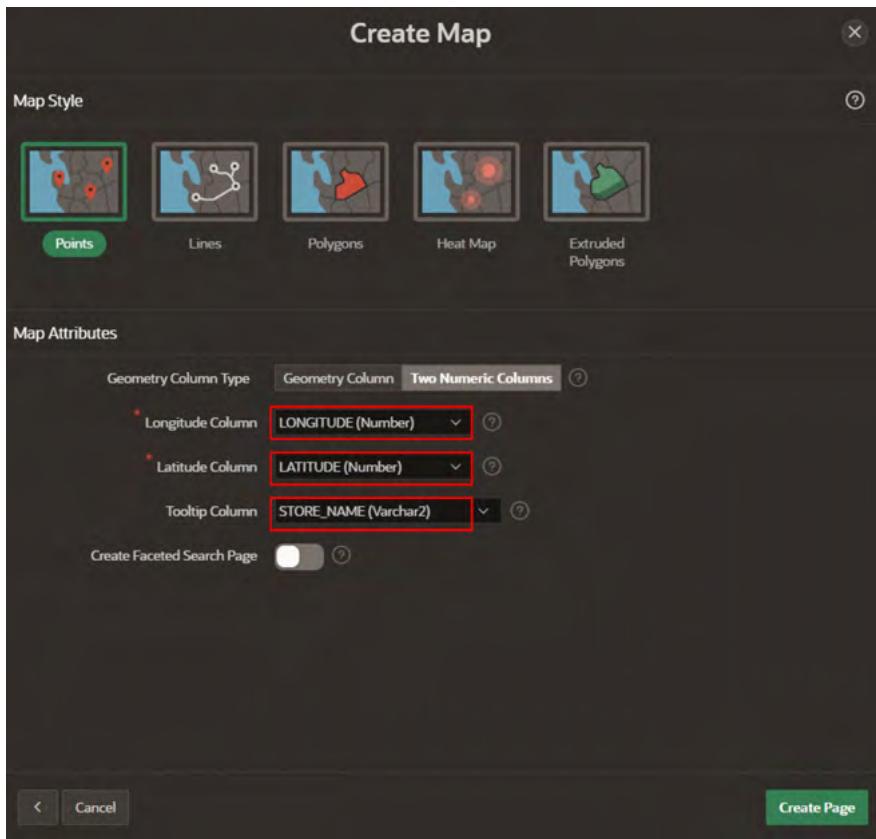


5. For **Create Map**, enter the following and click **Create Page**. For **Map Style**, select **Points**.

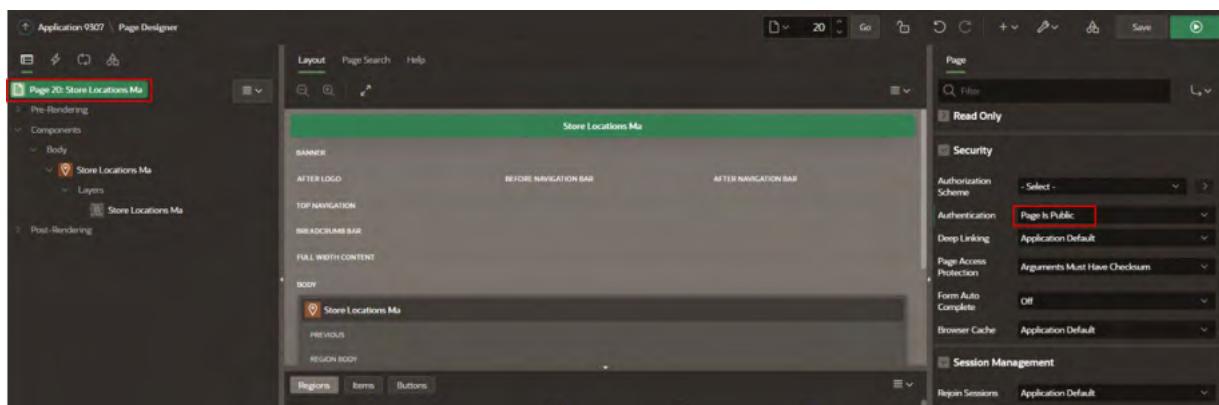
Under Map Attributes:

- For **Geometry column Type**, select **Two Numeric Columns**.
- For **Longitude Column**, select **LONGITUDE**.

- For **Latitude Column**, select **LATITUDE**.
- For **Tooltip Column**, select **STORE_NAME**.

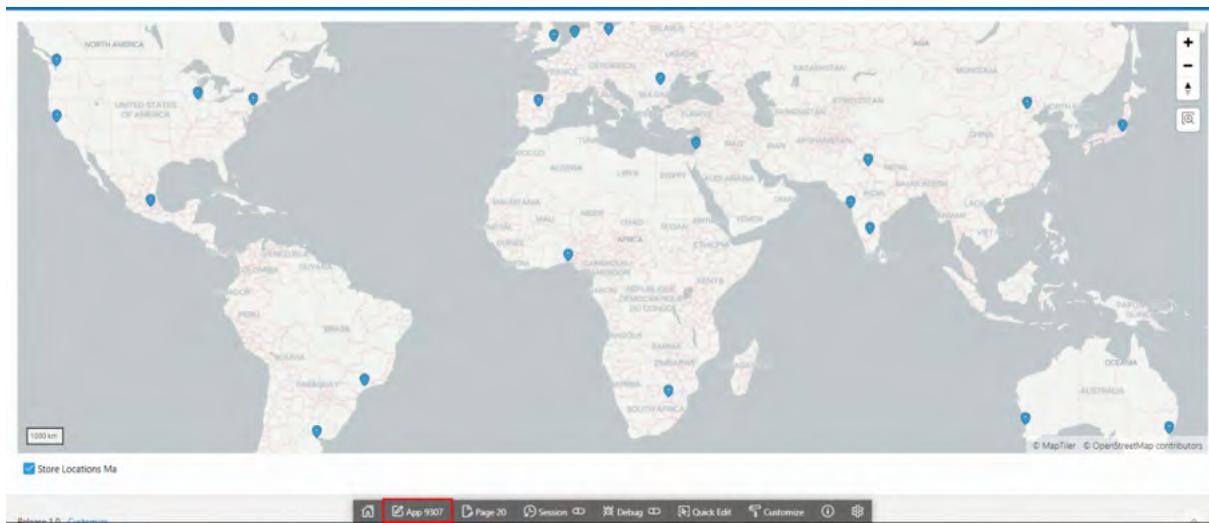


6. The Store Locations Map should be visible to the Public. To set the page as Public, select **Page #: Store Locations Map** in the Rendering tree. In the Property Editor, navigate to **Security**, and for **Authentication**, select **Page is Public**.

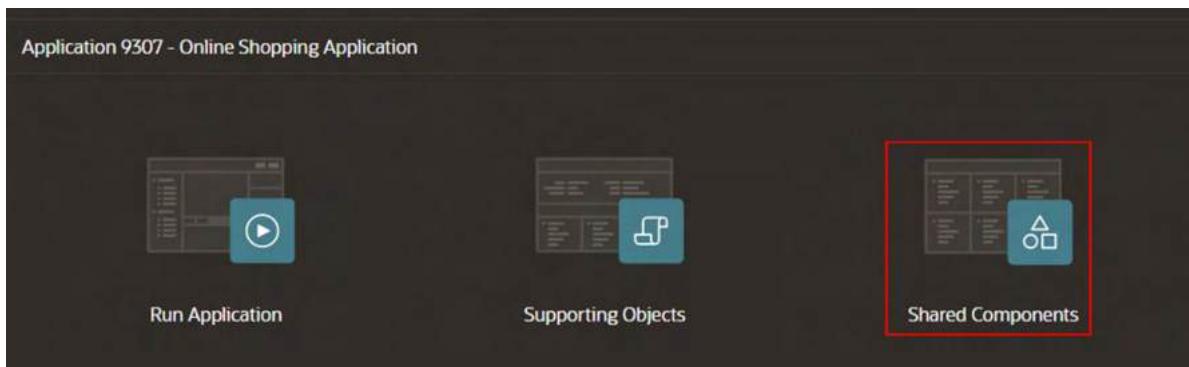


7. Then, click **Save** and **Run Page**.

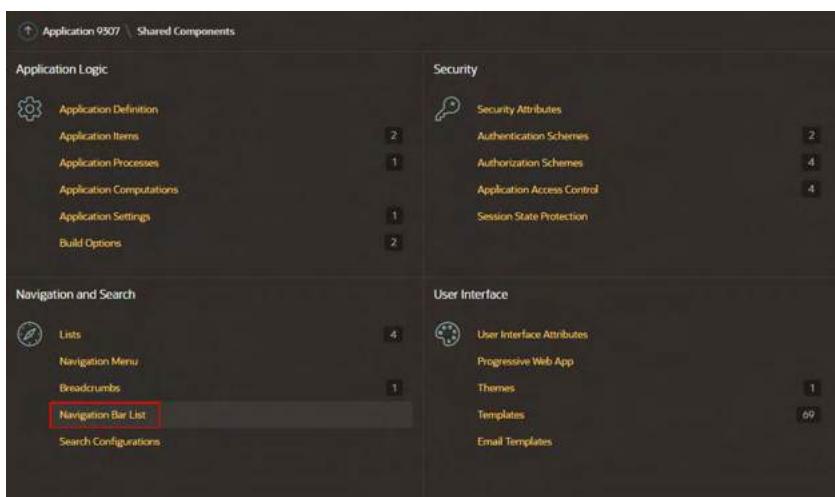
- The **Store Details Map** Page is now displayed. Now, in the developer toolbar, select **App < n >**.



- Navigate to **Shared Components**.



- In the **Shared Components** page, under **Navigation**, select **Navigation Bar List**.



11. Select Navigation Bar, under Lists.

The screenshot shows the 'Lists' tab selected in the top navigation bar. Below it is a search bar and a toolbar with 'Go', 'Actions', 'Reset', 'Copy', and 'Create' buttons. The main area displays a table titled 'Navigation Bar List'. The columns are: Name, Type, Entries, References, Updated, Navigation Bar, Navigation Menu, Subscribed From, and Subscribers. A single row is visible, labeled 'Navigation Bar' with 'Static' type, 0 entries, 1 reference, updated 3 days ago, and 'Yes' in the Navigation Bar column. The 'Navigation Bar' column is highlighted with a red border.

| Name | Type | Entries | References | Updated | Navigation Bar | Navigation Menu | Subscribed From | Subscribers |
|----------------|--------|---------|------------|------------|----------------|-----------------|-----------------|-------------|
| Navigation Bar | Static | 0 | 1 | 3 days ago | Yes | No | - | - |

12. Click Create Entry.

The screenshot shows the 'List: Navigation Bar' entry creation page. At the top right are 'Cancel', 'Delete', and 'Apply Changes' buttons. Below is a 'Name' field with 'Navigation Bar' entered. In the 'List Entries' section, there is a 'Create Entry >' button highlighted with a red box. The bottom part of the screen shows a table with columns: Sequence, Name, Target, Icon, Authorization Scheme, Build Option, Level, Parent Entry, Children, Conditional, and Updated On.

13. For List Entry, enter the following and click Create List Entry.

Under Entry:

- For List Entry Label, enter **Store Locations Map**.

Under Target:

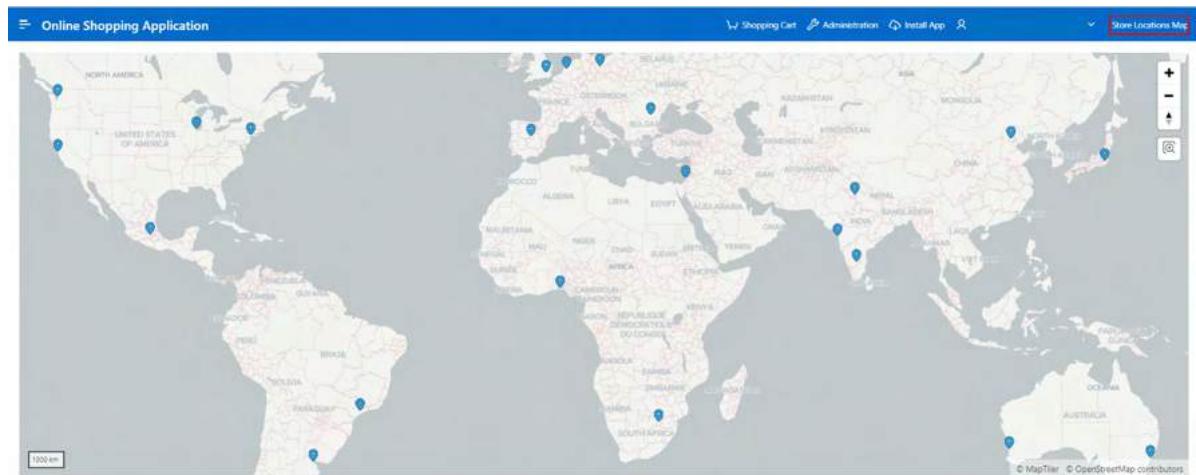
- For Page, select **20**.

The screenshot shows the 'List Entry' creation page. In the 'Entry' section, the 'List' is set to 'Navigation Bar'. The 'List Entry Label' field contains 'Store Locations Map', which is highlighted with a red box. In the 'Target' section, the 'Page' dropdown is set to '20', also highlighted with a red box. Other fields like 'Target type' and 'reset pagination for this page' are shown below.

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



15. You can now see that **Store Locations Map** is now displayed in **Navigation Bar**.



You now know how to manage Map pages. You may now **proceed to the next lab**.



Practice: Migrate Application Development Between Environments

Practice 1: Migrate Applications

Overview

In this lab, You will create Installation scripts and Data Packages to export Database objects along with Data. Then you will export the DEMO PROJECTS Application along with supporting objects and import it into another workspace.

Objectives

In this lab, you will:

- Create Installation Scripts
- Create Data Packages
- Export application with supporting Objects.

Downloads

- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Create Installation Scripts, Data packages, and Export Applications.

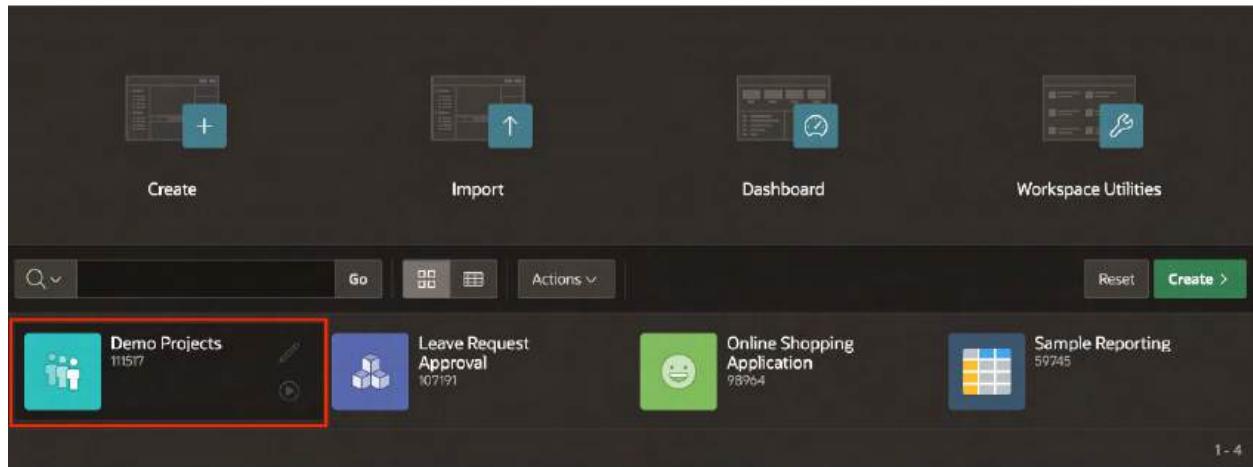
Before starting this lab, you need to provision a new **APEX instance** or **Workspace**.

In this hands-on lab, you will log in to a new, remote APEX workspace, create a table, and then **register schema with ORDS**.

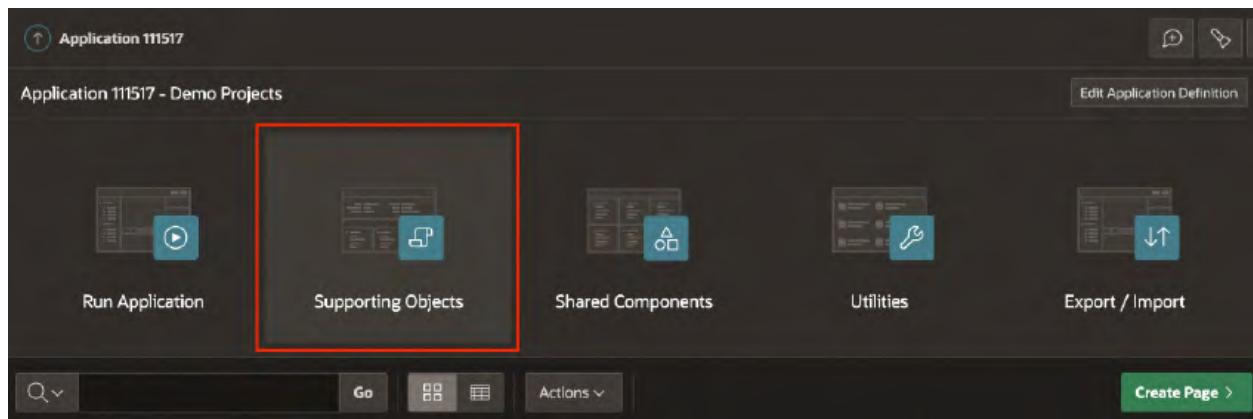
1. On the Workspace home page, click the App Builder icon.



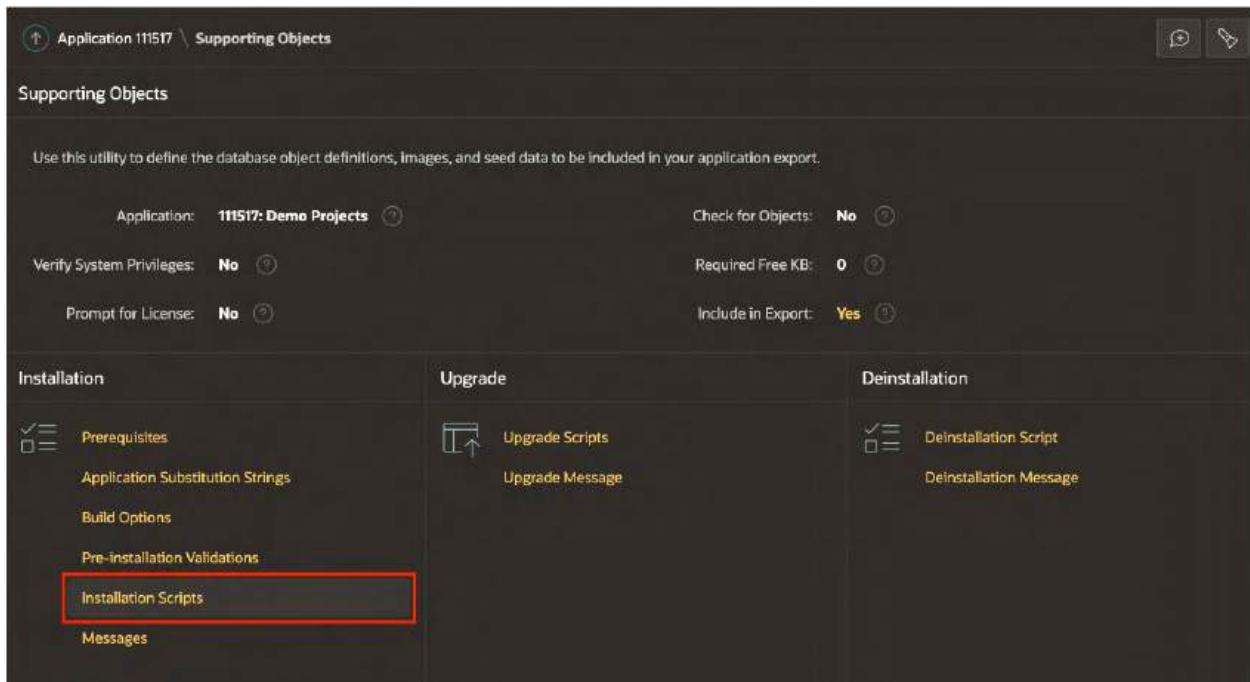
2. Select the **Demo Projects** application.



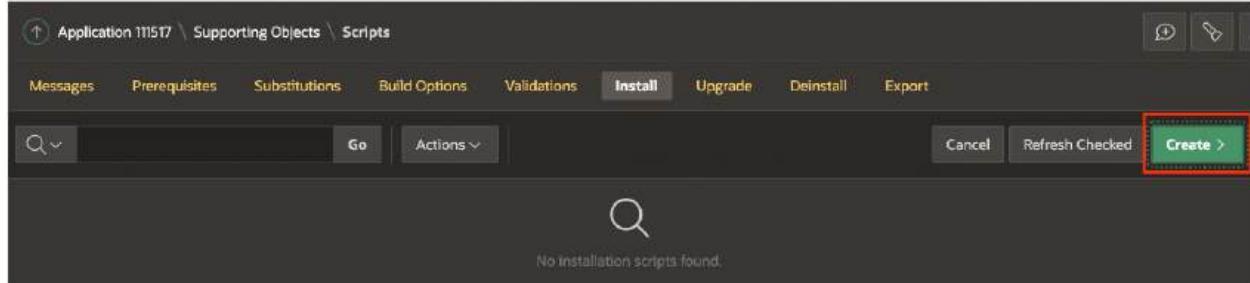
3. Click **Supporting Objects**. The Supporting Objects page appears.



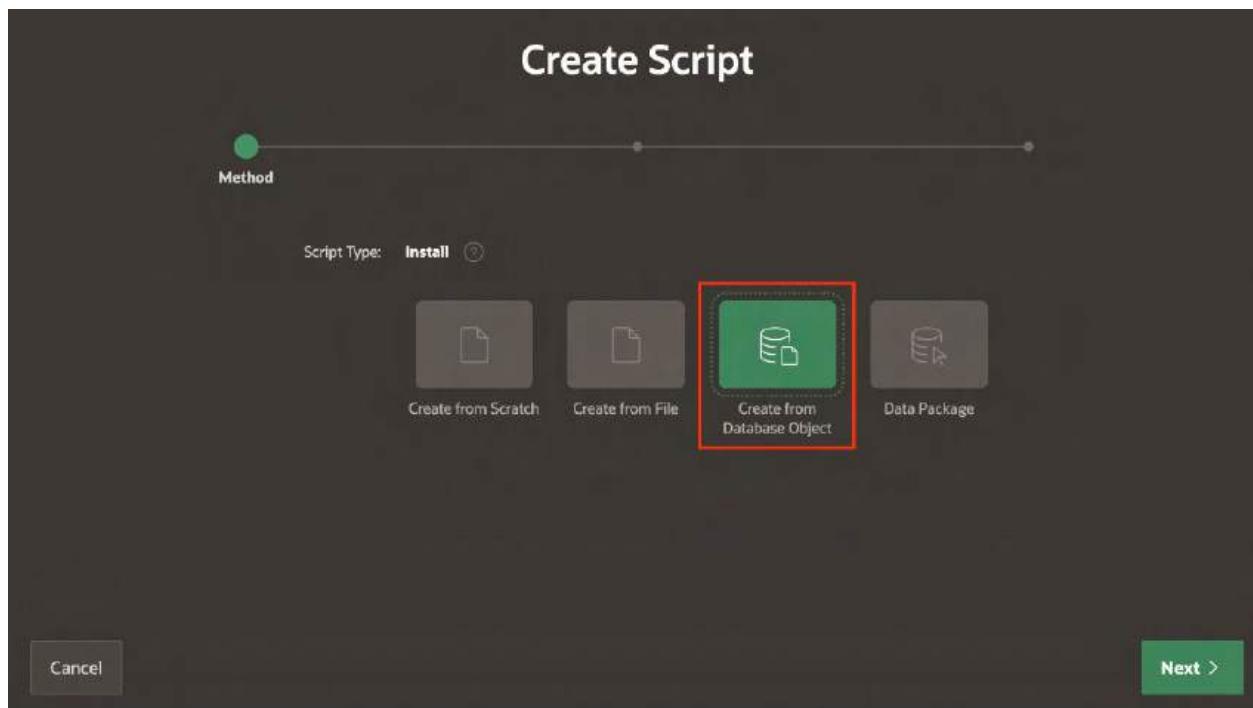
4. Under **Installation**, Select **Installation Scripts**.



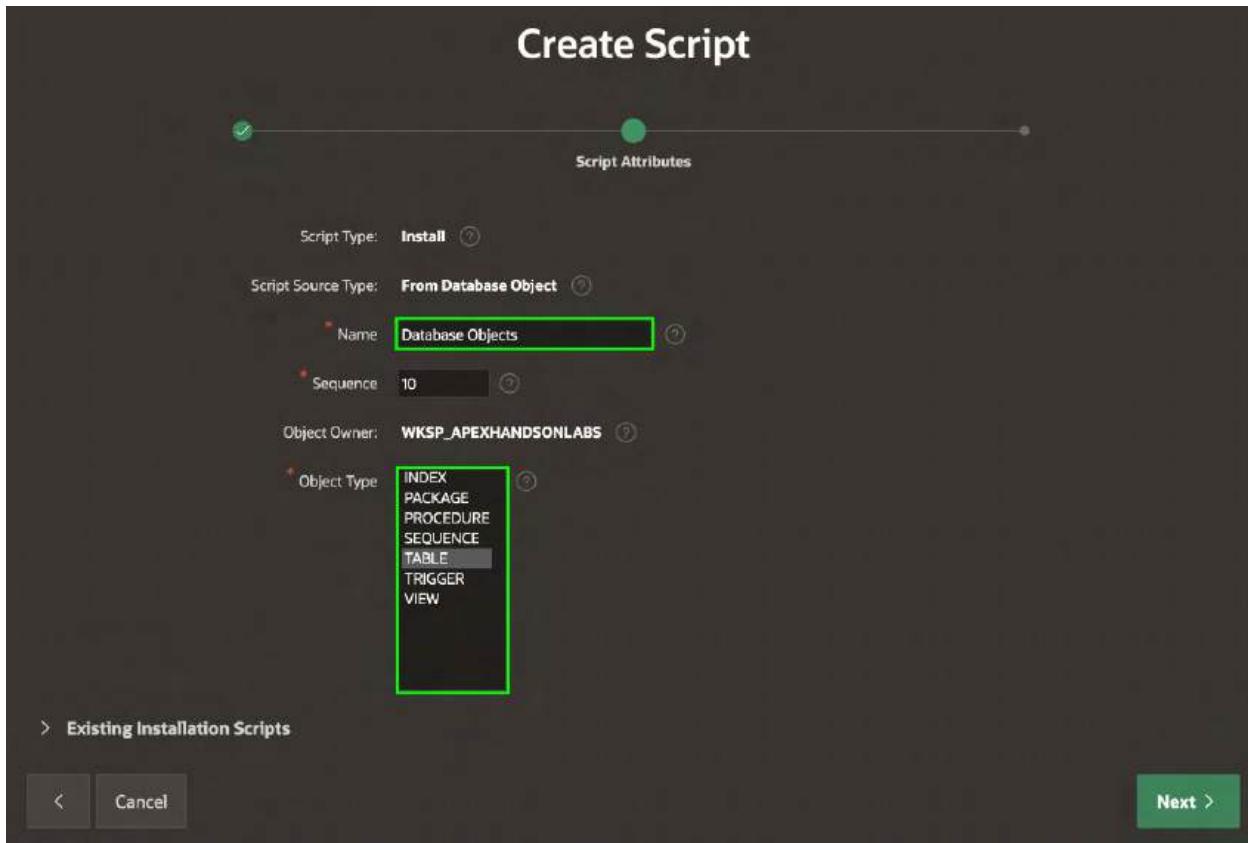
5. Click **Create**.



6. Select **Create from Database Object** under **Create Script**.



7. Under **Script Attributes**, select the below options and click **Next**.
- For **Name**, Select **Database Objects**.
 - For **Object Type**, select **Table****



8. Now set the following attributes and then click **Next**.
 - a. For **Table Options**, Select the **Checkbox**.
 - b. For **Object**, Select and Move the tables **DEMO_PROJECTS** and **EBA_DEMO_IR_EMP** to the right side.

Create Script

Warning

The script will be generated based on the underlying Oracle Database version, using version specific syntax. If this application is to be imported into earlier database versions you should test the script for backward compatibility. For example, if generating tables in Oracle Database 12c the script may contain IDENTITY Column clauses which will fail in Oracle Database 11g.

Script Type: **Install**

Script Source Type: **From Database Object**

Name: **Database Objects**

Sequence: **10**

Object Owner: **WKSP_APEXHANDSONLABS**

Object Types: **TABLE**

Filter Clear

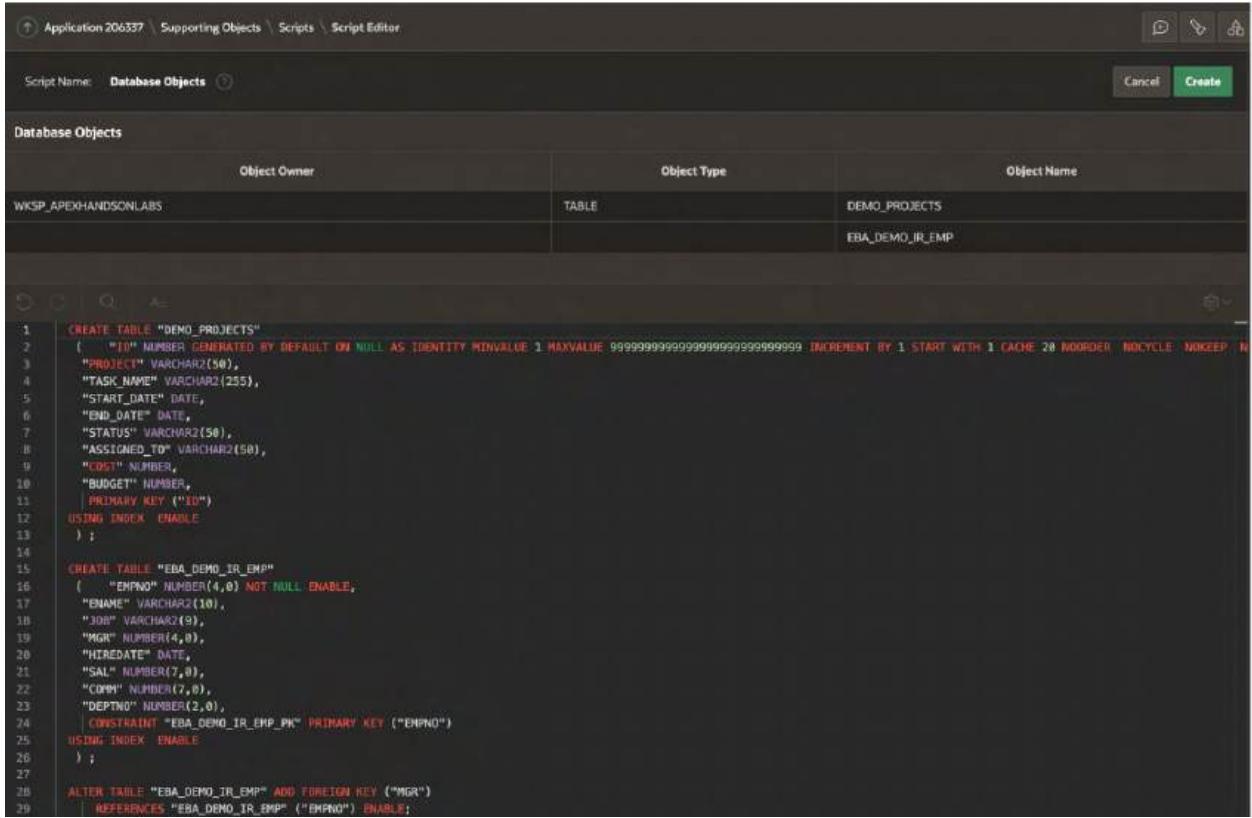
Object

- APX\$ERR\$DEMO_PROJECTS\$1!7200 (TABLE)
- APX\$ERR\$EBA_DEMO_IR_EMP\$5B5229 (TABLE)
- CLOTHING_LOOKUP (TABLE)
- COLOR_LOOKUP (TABLE)
- CUSTOMERS (TABLE)
- DEPARTMENTS (TABLE)
- DEPARTMENT_LOOKUP (TABLE)
- DEPT (TABLE)
- DR\$EBA_DEMO_JG_TEXT_FTX\$1 (TABLE)
- DR\$EBA_DEMO_JG_TEXT_FTX\$K (TABLE)
- DR\$EBA_DEMO_JG_TEXT_FTX\$N (TABLE)
- DR\$EBA_DEMO_JG_TEXT_FTX\$U (TABLE)
- EBA_DEMO_APPR_APPROVERS (TABLE)
- EBA_DEMO_APPR_DEPT (TABLE)
- EBA_DEMO_APPR_EMP (TABLE)

DEMOS_PROJECTS (TABLE)
EBA_DEMO_IR_EMP (TABLE)

< Cancel Next >

9. You will now see the **Scripts** created. Click **Create**.

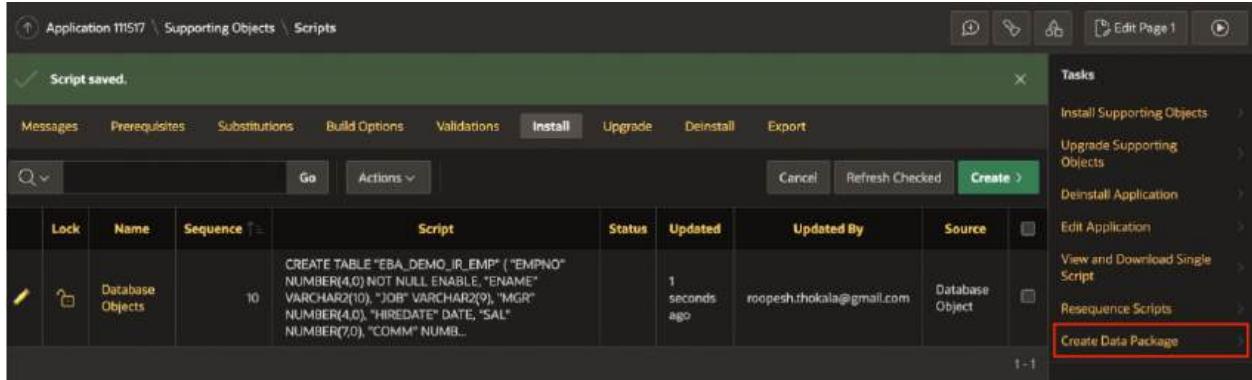


```

1 CREATE TABLE "DEMO_PROJECTS"
2 (
3     "ID" NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE NOKEEP N
4     "PROJECT_NAME" VARCHAR2(50),
5     "TASK_NAME" VARCHAR2(255),
6     "START_DATE" DATE,
7     "END_DATE" DATE,
8     "STATUS" VARCHAR2(50),
9     "ASSIGNED_TO" VARCHAR2(50),
10    "COST" NUMBER,
11    "BUDGET" NUMBER,
12    PRIMARY KEY ("ID")
13    USING INDEX ENABLE
14 );
15
16 CREATE TABLE "EBA_DEMO_IR_EMP"
17 (
18     "EMPNO" NUMBER(4,0) NOT NULL ENABLE,
19     "ENAME" VARCHAR2(10),
20     "JOB" VARCHAR2(9),
21     "MGR" NUMBER(4,0),
22     "HIREDATE" DATE,
23     "SAL" NUMBER(7,0),
24     "COMM" NUMBER(7,0),
25     "DEPTNO" NUMBER(2,0),
26     CONSTRAINT "EBA_DEMO_IR_EMP_PK" PRIMARY KEY ("EMPNO")
27     USING INDEX ENABLE
28 );
29
30 ALTER TABLE "EBA_DEMO_IR_EMP" ADD FOREIGN KEY ("MGR")
31     REFERENCES "EBA_DEMO_IR_EMP" ("EMPNO") ENABLE;

```

10. Now, you will see that the Scripts got created and Saved. You will need to include **Data** in your **Supporting Objects**. Select **Create Data Package** under **Tasks** on the Right side of your page.



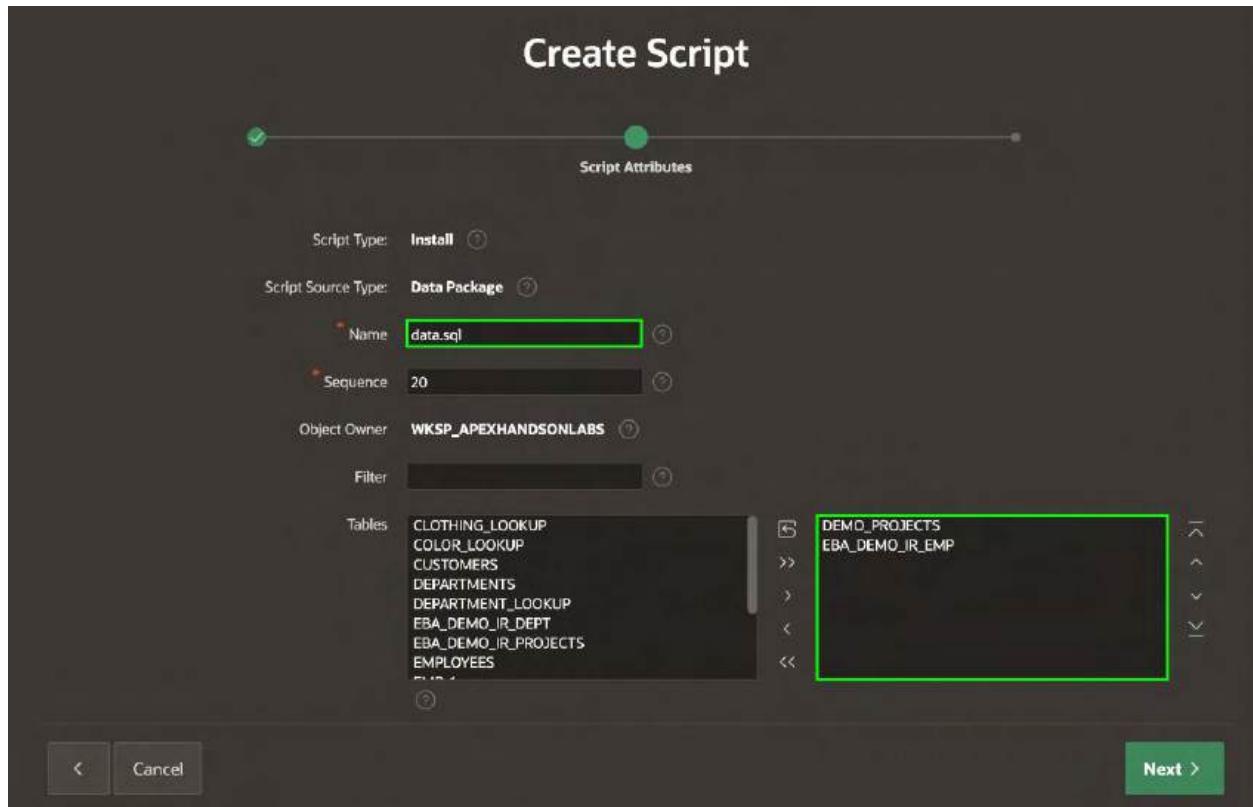
| Lock | Name | Sequence | Script | Status | Updated | Updated By | Source |
|------|------------------|----------|---|--------|---------------|---------------------------|-----------------|
| | Database Objects | 10 | CREATE TABLE "EBA_DEMO_IR_EMP" ("EMPNO" NUMBER(4,0) NOT NULL ENABLE, "ENAME" VARCHAR2(10), "JOB" VARCHAR2(9), "MGR" NUMBER(4,0), "HIREDATE" DATE, "SAL" NUMBER(7,0), "COMM" NUMB... | | 1 seconds ago | roopesh.thokala@gmail.com | Database Object |

Create >

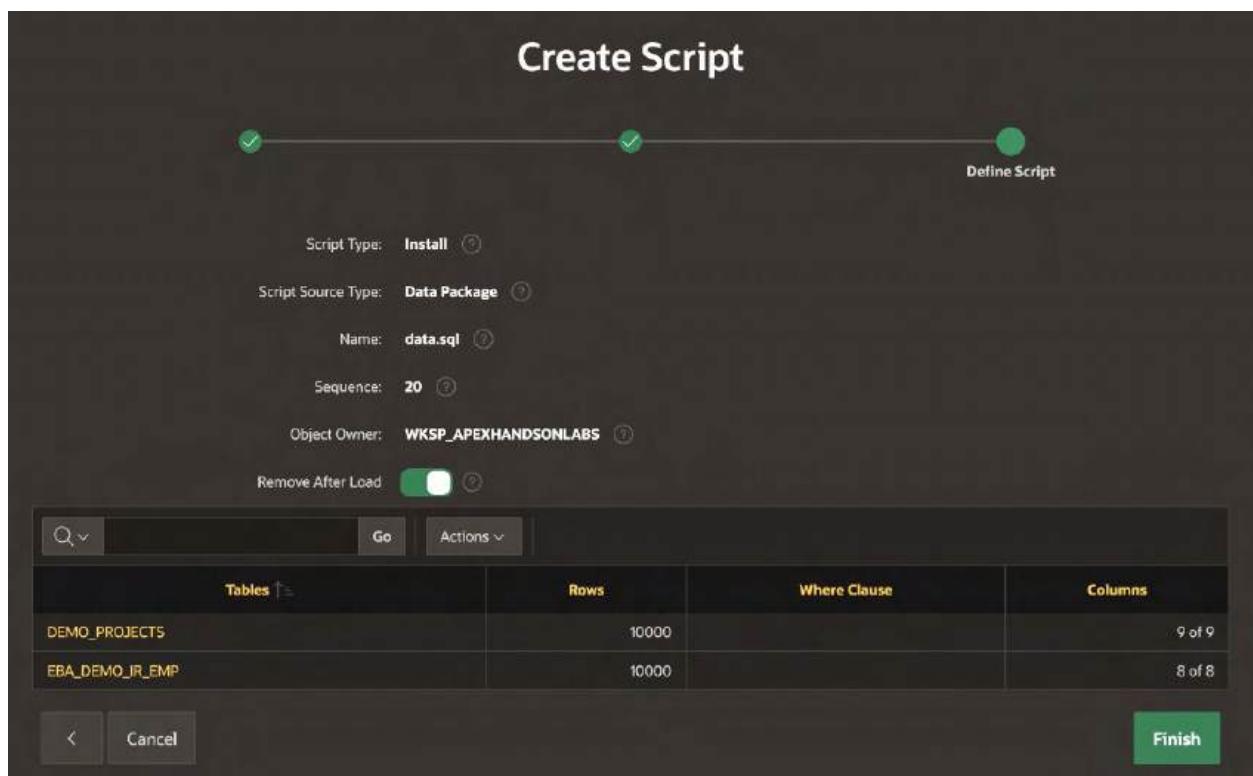
Create Data Package

11. Under **Script Attributes**, Select the following and click **Next**.

- For **Name**, select **data.sql**
- For **Tables**, ensure **DEMO_PROJECTS** and **EBA_DEMO_IR_EMP** are listed on the right side, and if not, select and shuttle these two tables over there.



12. For **Create Script**, accept the defaults and click **Finish**.



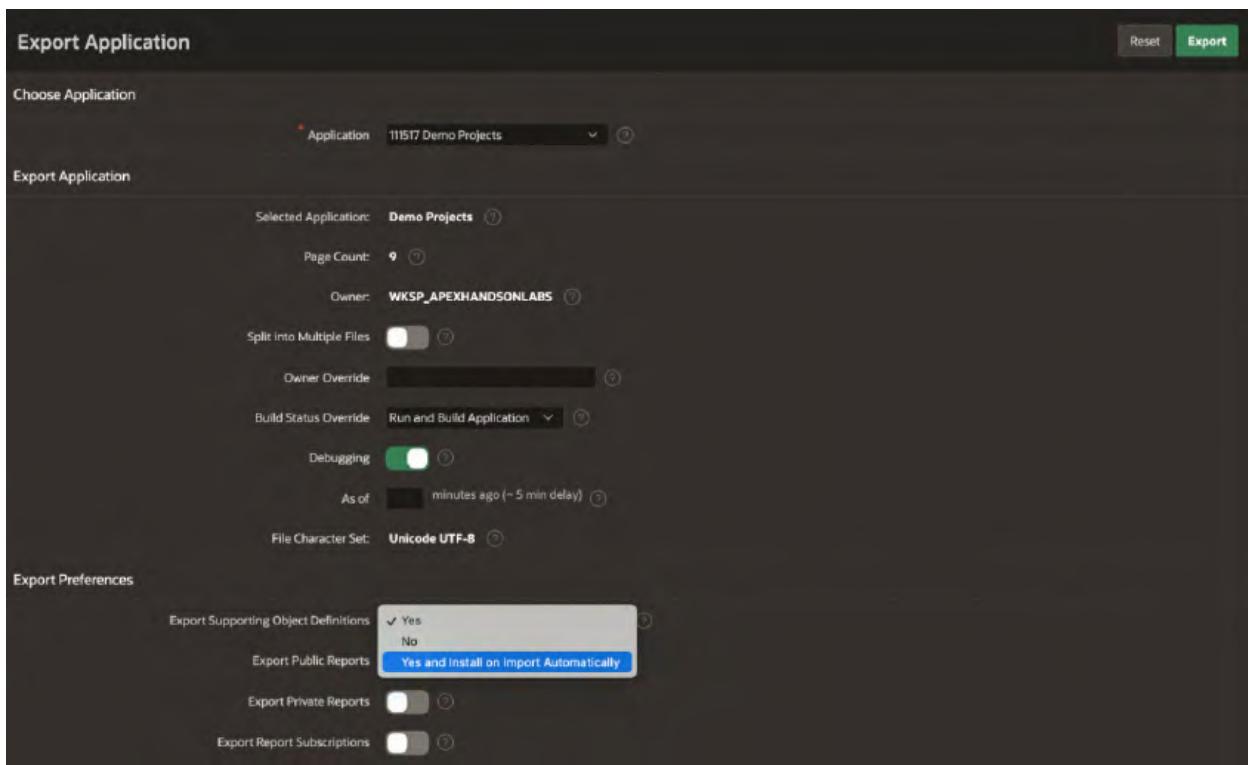
13. The Scripts are now Created. Navigate back to the application and Select **Export/Import**.

The screenshot shows two windows side-by-side. The top window is titled 'Application 111517 \ Supporting Objects \ Scripts'. It displays a table of scripts with columns: Lock, Name, Sequence, Script, Status, Updated, Updated By, and Source. There are two entries: one for 'Database Objects' (Sequence 10) and one for 'data.sql' (Sequence 20). The bottom window is titled 'Application 111517 - Demo Projects'. It shows a navigation bar with icons for Run Application, Supporting Objects, Shared Components, Utilities, and Export / Import (which is highlighted with a red box). Below the navigation bar is a search bar and a 'Create Page >' button.

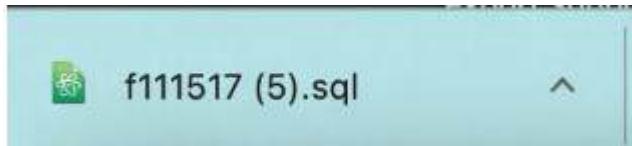
14. Select **Export** and click **Next**.

This is a modal dialog box with the title 'What task would you like to perform?'. It contains three options: 'Import' (with an upward arrow icon), 'Export' (with a downward arrow icon, which is highlighted with a red box), and 'Remote Deployment' (with a cloud icon). At the bottom are 'Cancel', 'Help', and 'Next >' buttons.

15. In the **Export Application** page, Under **Export Preferences**, select **Yes and Install on Import Automatically** and Click **Export**.

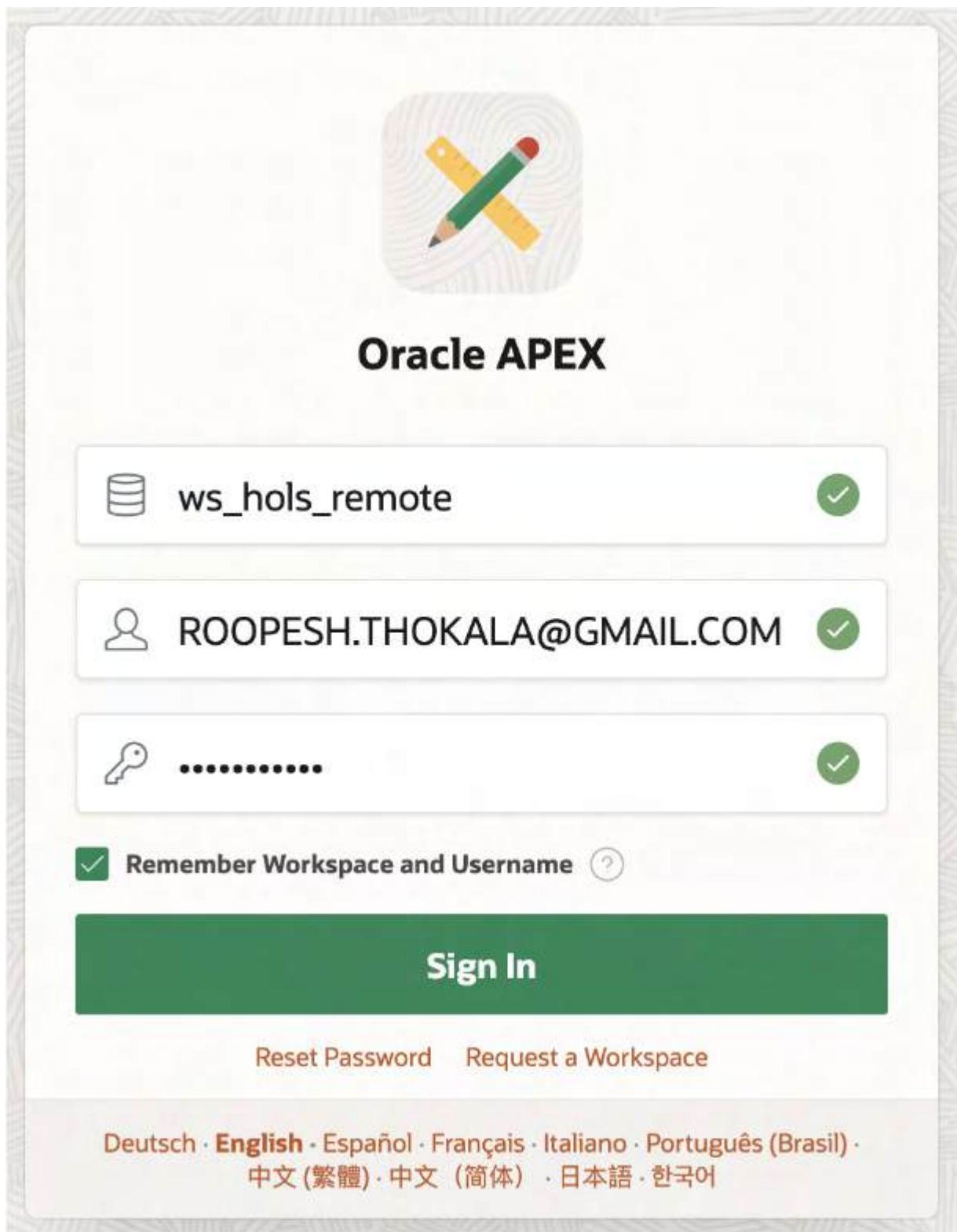


16. You will now notice that the application is downloaded as **fNNN.sql**

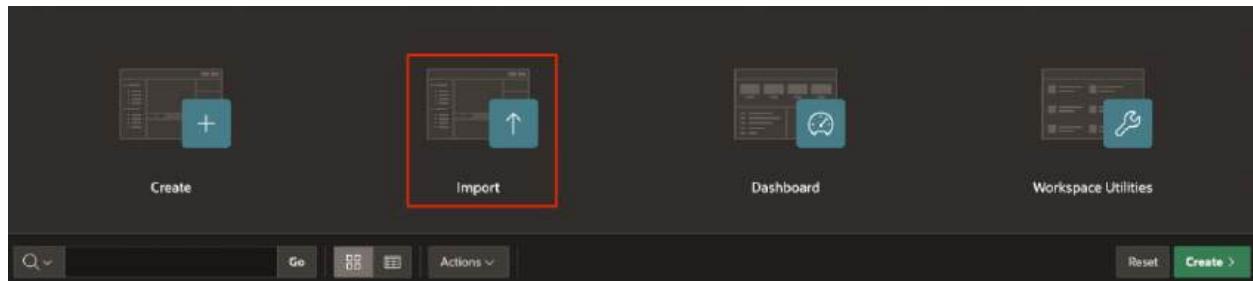
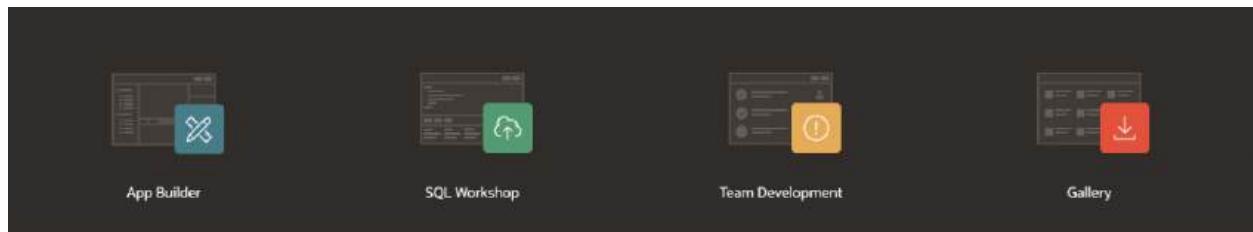


Import Application into the Target Workspace and Verify Database Objects.

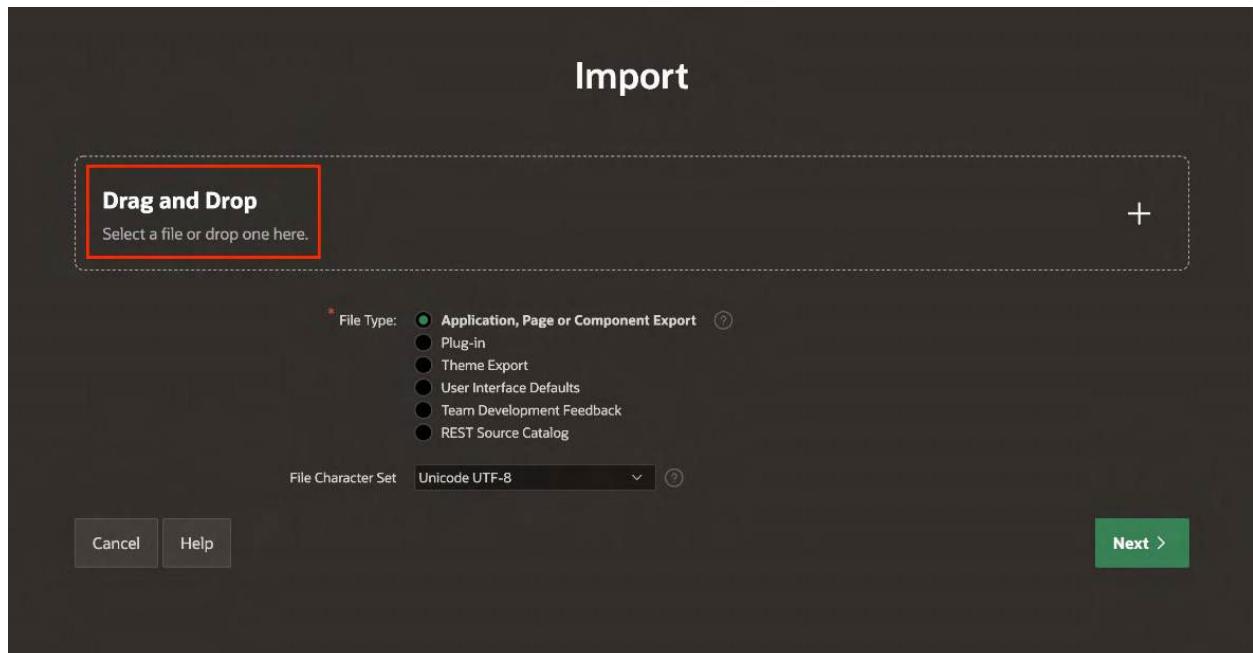
1. Log in to the Target **Workspace**.



2. Click **App Builder** and then select **Import**.



3. Under **Import**, click **Drag and Drop**.

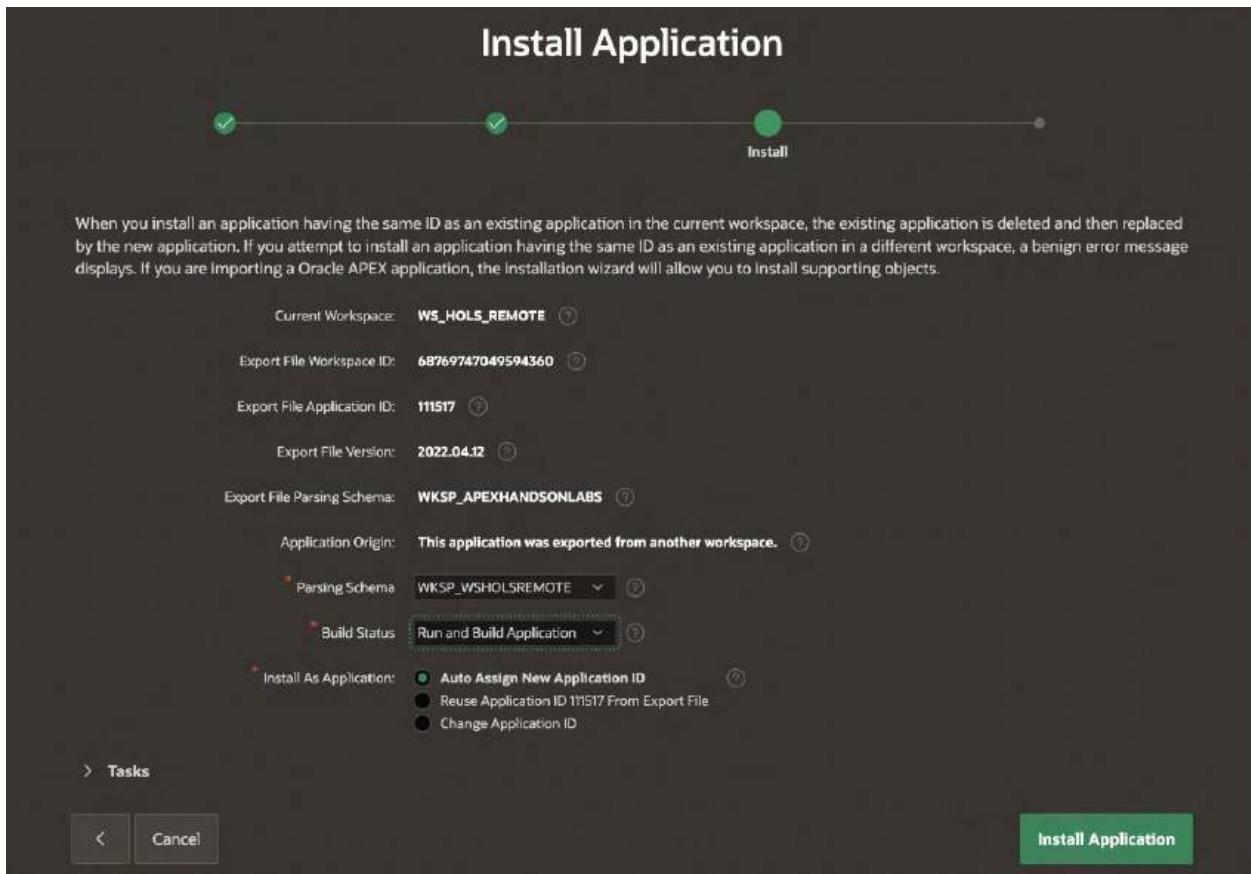


4. Choose the file we exported in the previous lab from your **Local System** and then click **Next**.

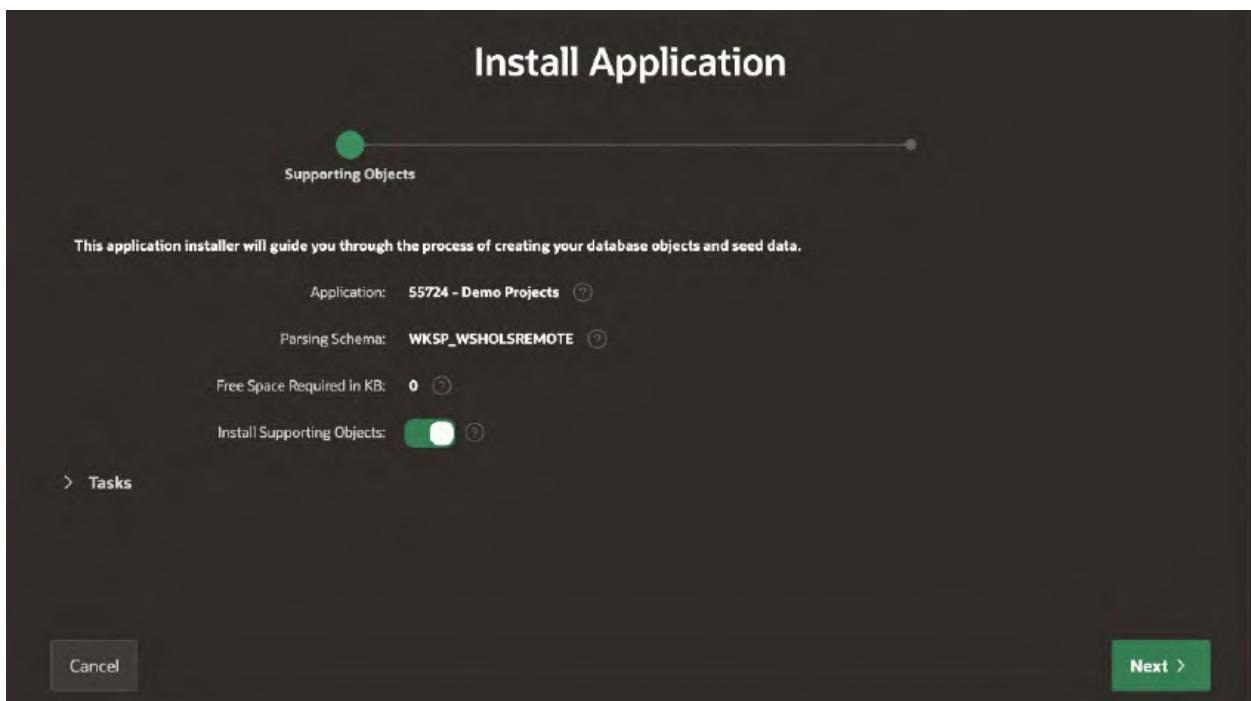
The screenshot shows a file browser window with the following details:

- Favourites:** Recents, Applications, Downloads.
- Downloads folder:** Name: f111517 (5).sql, Size: 642 KB, Kind: SQL source.
- Import Confirmation Screen:** Title: Import. Progress bar: Step 1 of 3 completed. Message: "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." A "Tasks" link is visible.
- Buttons:** Back, Cancel, Next >.

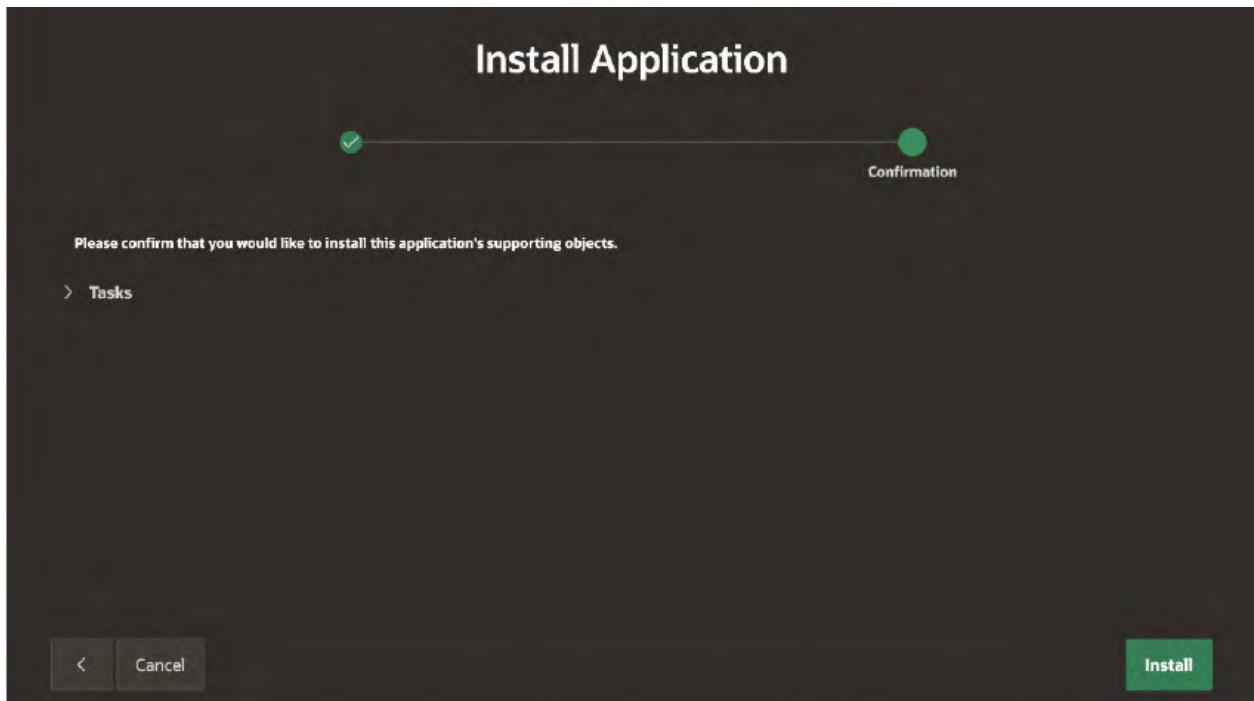
5. Under **Install**, Set **Build Status** to **Run and Build Application**. Leave the remaining settings to default and Click **Install Application**.



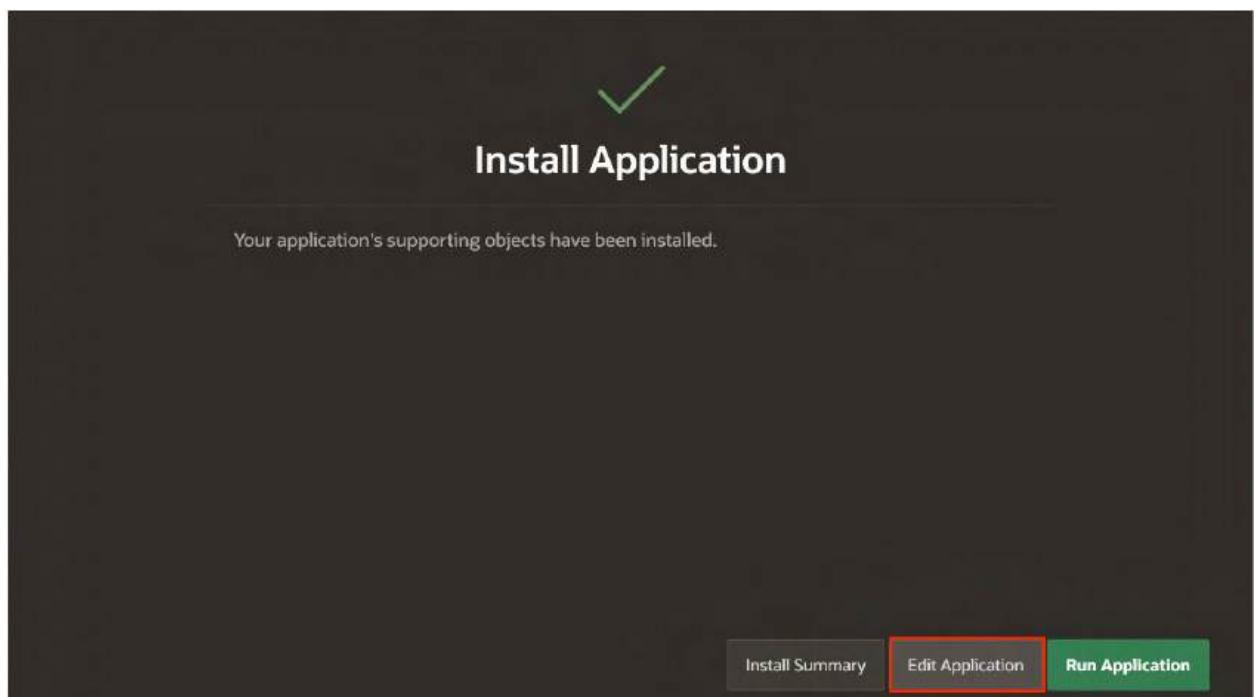
6. For **Supporting Objects**, Set **Install Supporting Objects** to **yes** and then click **Next**.



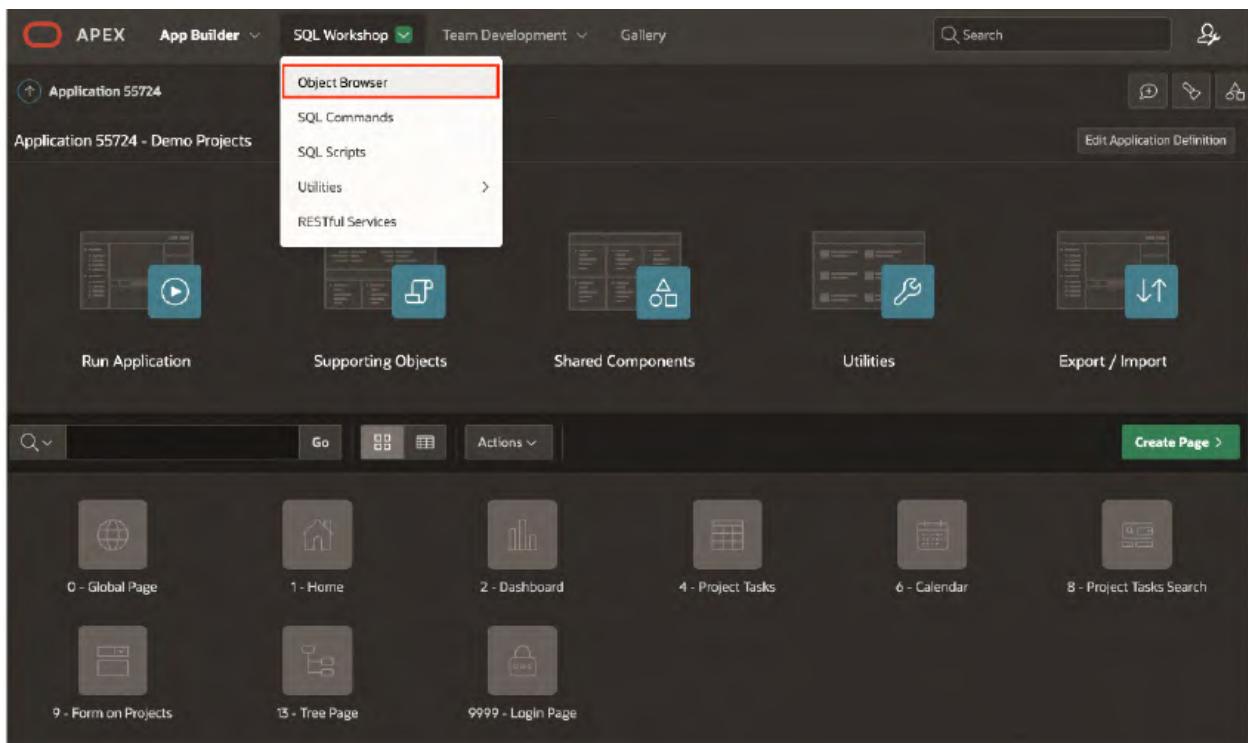
7. Click **Install**.



8. You will now see that **Your application's supporting objects have been installed**.
Click **Edit Application** to review the **Database Objects**.



9. Click on the Arrow mark next to **SQL Workshop** and then click **Object Browser**.



10. Select **DEMO_PROJECTS** table and then click **Data**. Verify the Data in the table **DEMO_PROJECTS**.

The screenshot shows the Oracle APEX Object Browser interface. On the left, there's a sidebar with categories like Tables, Views, Indexes, Sequences, Types, Packages, Procedures, Functions, Triggers, Database Links, Materialized Views, Synonyms, and SODA Collections. Under the 'Tables' section, the 'DEMO_PROJECTS' table is selected and highlighted with a red box. The main panel shows the 'DEMO_PROJECTS' table structure with columns: ID, PROJECT, TASK_NAME, START_DATE, END_DATE, STATUS, ASSIGNED_TO, COST, and BUDGET. There are 14 rows of data listed, each with edit icons. The 'Data' tab is selected at the top of the table view. The bottom of the screen shows copyright information: Copyright © 1999-2023, Oracle and/or its affiliates, Oracle APEX 23.1.0.19, and a footer note: 1-20.

| | ID | PROJECT | TASK_NAME | START_DATE | END_DATE | STATUS | ASSIGNED_TO | COST | BUDGET |
|----|----|--------------------|----------------------|------------|------------|---------|---------------|------|--------|
| 1 | 1 | ACME Web Con... | Identify server r... | 11/19/2021 | 12/06/2021 | Closed | John Watson | 500 | 300 |
| 2 | 2 | Maintain Suppor... | HR software upg... | 12/04/2021 | 01/01/2022 | On-Hold | Pam King | 8000 | 9000 |
| 3 | 3 | Maintain Suppor... | Apply Billing Sys... | 01/03/2022 | 01/29/2022 | On-Hold | Russ Sanders | 9500 | 2000 |
| 4 | 4 | ACME Web Con... | Determine Web L... | 12/16/2021 | 12/16/2021 | Closed | James Cassidy | 100 | 100 |
| 5 | 5 | ACME Web Con... | Specify security ... | 01/02/2022 | 01/04/2022 | Closed | John Watson | 200 | 300 |
| 6 | 6 | ACME Web Con... | Select servers fo... | 11/24/2021 | 11/26/2021 | Closed | James Cassidy | 200 | 600 |
| 7 | 7 | Email Integration | Complete plan | 11/17/2021 | 12/15/2021 | Closed | Mark Nile | 3000 | 1500 |
| 8 | 8 | ACME Web Con... | Configure Works... | 12/20/2021 | 01/04/2022 | Closed | John Watson | 200 | 100 |
| 9 | 9 | ACME Web Con... | Create pilot wor... | 01/18/2022 | 02/12/2022 | Closed | John Watson | 100 | 100 |
| 10 | 10 | ACME Web Con... | Run Installation | 11/22/2021 | 12/12/2021 | Closed | James Cassidy | 100 | 100 |
| 11 | 11 | Bug Tracker | Implement bug t... | 01/05/2022 | 01/10/2022 | Closed | Myra Sutcliff | 100 | 100 |
| 12 | 12 | Bug Tracker | Review automati... | 11/25/2021 | 11/29/2021 | On-Hold | Myra Sutcliff | 2750 | 1500 |
| 13 | 13 | Train Developers | Publish develop... | 12/03/2021 | 12/15/2021 | On-Hold | John Watson | 1000 | 2000 |
| 14 | 14 | Train Developers | Publish links to ... | 12/28/2021 | 01/05/2022 | Closed | John Watson | 100 | 100 |

Practice 2: Remote Deployment

Overview

In this lab, You will be using Remote Deployment to deploy your Application from Development to UAT. The ORDS feature, **REST Enabled SQL** is not available for **apex.oracle.com** users. Therefore, you will not be able to perform this lab in the hosted instance apex.oracle.com. You should be using **APEX Service** or **APEX on Autonomous Database** options for this lab.

Objectives

In this lab, you will:

- Export application with supporting Objects.
- Use Remote Deployment to deploy your Application.

Downloads

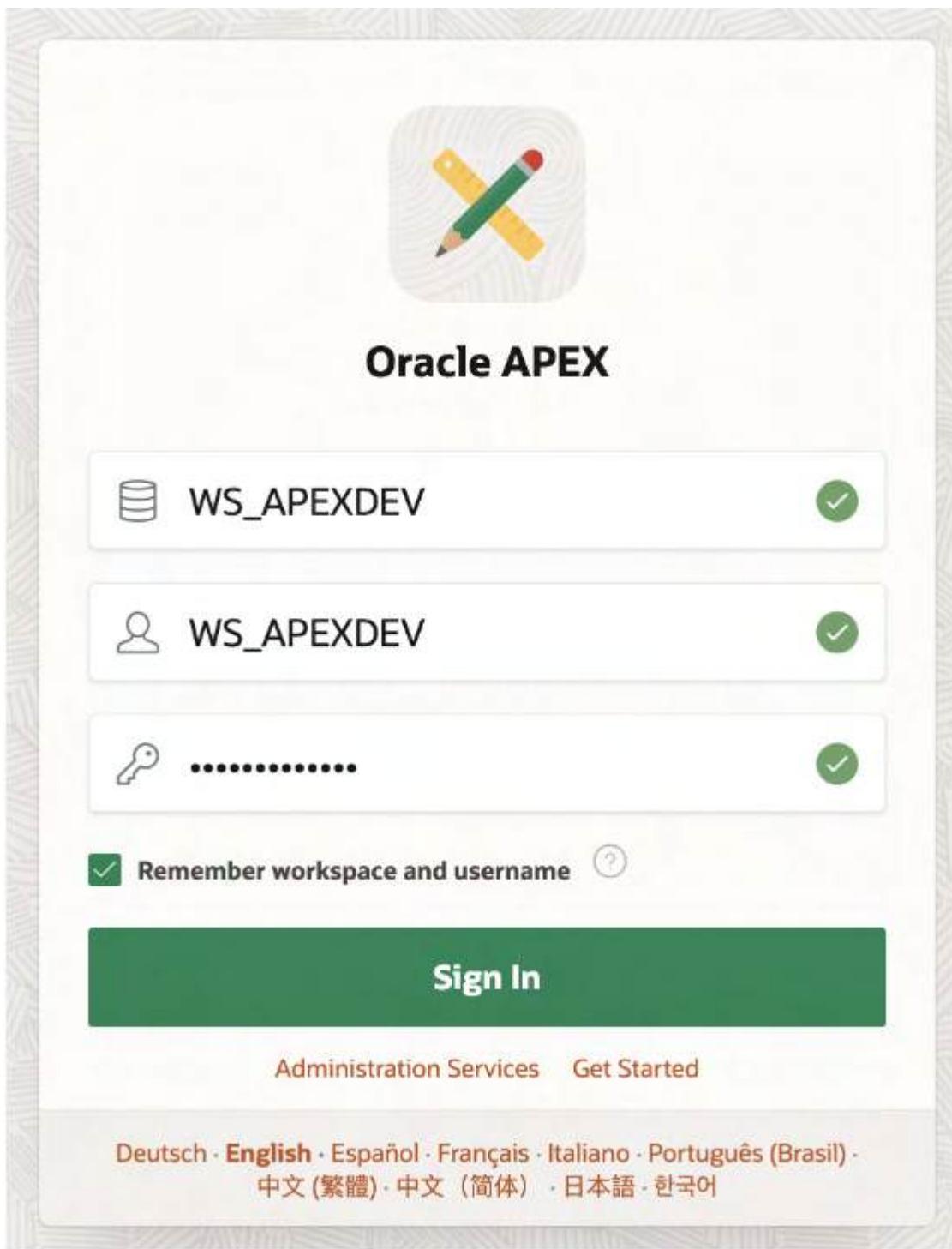
- Did you miss out on trying the previous labs? Don't worry! You can download the application from [here](#) and import it into your workspace. To run the app, please run the steps described in [Get Started with Oracle APEX](#) and [Using SQL Workshop](#) workshops.

Import Sample Application

The ORDS feature, **REST Enabled SQL** is not available for **apex.oracle.com** users. Therefore, you will not be able to perform this lab in the hosted instance apex.oracle.com. You should be using **APEX Service** or **APEX on Autonomous Database** options for this lab.

In this Lab, You will install a **Sample Application** in a new workspace you created either on **APEX Service** or **APEX on Autonomous Database**. Please follow the steps in **Get Started with Oracle APEX** workshop to provision an instance or to create workspace on **APEX Service** or **APEX on Autonomous Database**.

1. Log in to the new Workspace you created.



2. Once you Navigate to Gallery, Click **Sample Apps**.

| Application | Description | Version | More Info | Install |
|---------------------------------|---|---------|---------------------------|-------------------------|
| Sample Approvals | This application highlights the native Approval Components capabilities in Oracle APEX. It allows users to manage changes to employees' salaries and jobs after getting the approval of an appropriate individual, highlighting the key features of the Oracle APEX Approval components. | 22.1.0 | More Info | Install |
| Sample Calendar | This application highlights the native calendaring capabilities of Oracle APEX. It features a monthly calendar with stylized daily tasks. The dates can be changed using drag and drop, which is all declarative and easily created using native APEX wizards. | 22.1.0 | More Info | Install |
| Sample Cards | This application highlights Cards regions in Oracle APEX. Cards regions are a native region type. They provide developers with a powerful and flexible new way to display data in bite-sized blocks, ideal for use in faceted search, or presenting at-a-glance information. | 22.1.0 | More Info | Install |
| Sample Charts | This application highlights the charting capabilities of Oracle APEX. It demonstrates how you can enhance your applications to visually represent your data, using declarative and plug-in-based charting solutions. | 22.1.0 | More Info | Install |
| Sample Collections | Sample Collections enables you to store rows of data for use within an Oracle APEX session. This database application illustrates how to use PL/SQL to create and manage collection-based session state. | 22.1.0 | More Info | Install |
| Sample Data Loading | This application is built on simple EMP and DEPT tables to highlight how developers can define pages to allow end users to upload spreadsheet data into an existing table. | 22.1.0 | More Info | Install |
| Sample Dynamic Actions | This application demonstrates a number of different dynamic actions that can be incorporated into an application. These declarative client-side behaviors include simple examples for manipulating the display of components, style examples for changing the appearance of components, and server-side examples which interact with the database. | 22.1.0 | More Info | Install |
| Sample File Upload and Download | Learn how to create Oracle APEX applications that include file upload and download. Upload files using dialogs as well as dedicated pages. See how to download files stored in Oracle database BLOB columns within database tables. Specifically see how to produce file download links in interactive reports, classic reports, forms, and dynamically created HTML content. | 22.1.0 | More Info | Install |

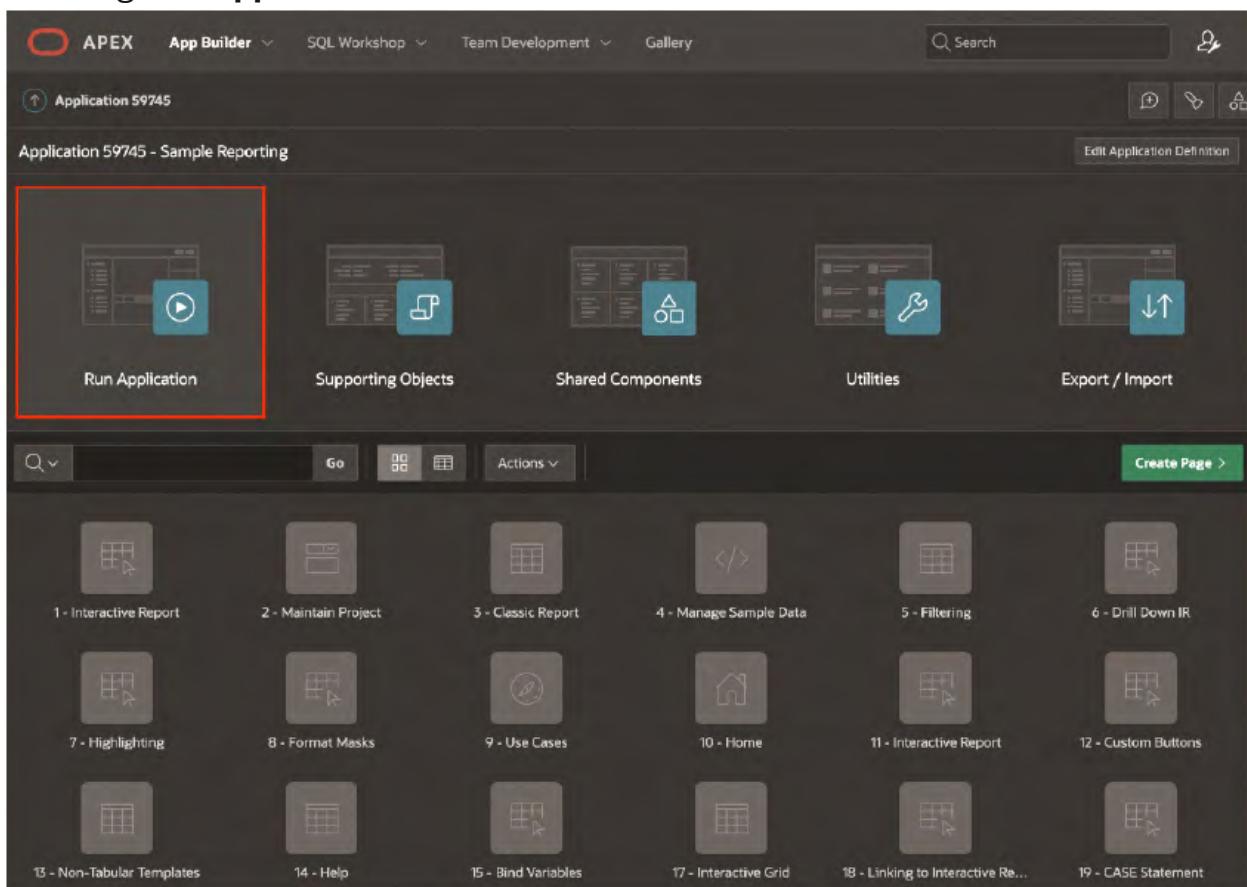
3. Now that you are in **Sample Apps**, search for apps by typing "**Reporting**" into the "Search Apps" search box and navigate to **Sample Reporting** and click the **Install** Button.

| Application | Description | Version | More Info | Install |
|---------------------------|--|---------|---------------------------|-------------------------|
| Sample Maps | This application contains numerous examples of visualizing coordinate data on a map. Use Map Markers, Lines or Polygons, or the Heat Map feature. The APEX Map Region can easily be combined with Oracle Spatial functionality (which is included in every Oracle Database) to perform a 'Within Distance Search', 'Nearest Neighbor Search', or other spatial analysis. | 22.1.0 | More Info | Install |
| Sample Master Detail | This application highlights the native master detail capabilities of Oracle APEX. The application contains four different master detail page layouts. The first two layouts display master detail in a single page using editable Interactive Grids. The last two layouts display master detail in two pages with mix of editable Interactive Grids, form items, classic reports and modal popups. | 22.1.0 | More Info | Install |
| Sample Reporting | This application highlights the reporting capabilities of Oracle APEX. You can create Interactive Reports, Interactive Grids, Faceted Search Reports, Cards Reports, and Classic Reports declaratively using SQL. | 22.1.0 | More Info | Install |
| Sample REST Services | This application showcases how to access external REST services from Oracle APEX. The app works on the sample RESTful Service, oracle.example.hr. The examples in this application illustrate how to create a simple tabular report on REST service data, how to filter, and how to add pagination. | 22.1.0 | More Info | Install |
| Sample Trees | Learn how to create a tree control using a SQL query. This application shows various methods of integrating tree controls into your Oracle APEX application. | 22.1.0 | More Info | Install |
| Universal Theme Reference | This app introduces you to Universal Theme by providing an easy way to browse through the various templates, template options, and theme styles. The examples demonstrate how you can easily control the layout of your pages, to create a great looking application. | 22.1.0 | More Info | Install |

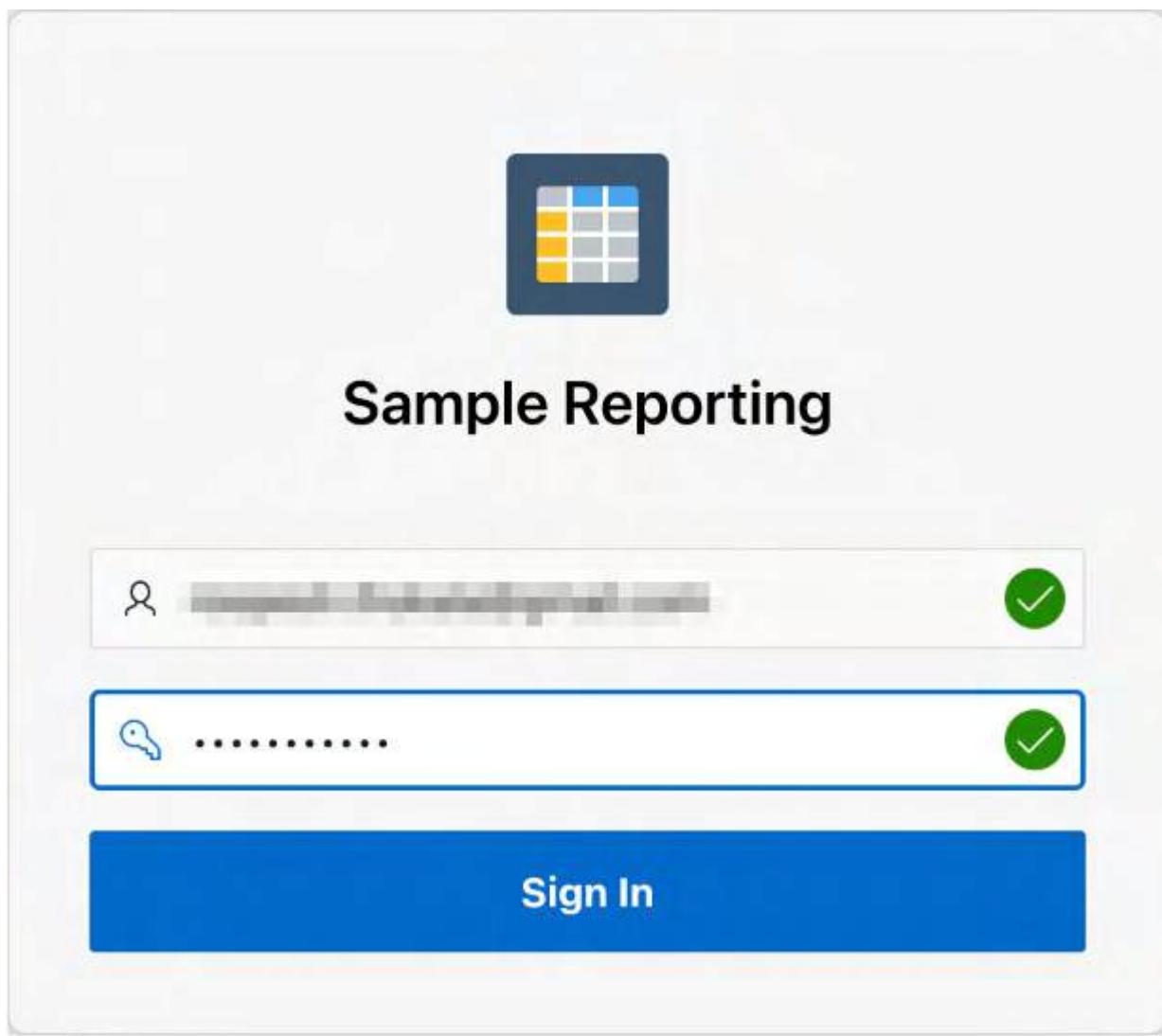
4. On the Install Application page, click the **Install Application** Button.



5. You can now see that the application is installed. Run the application by selecting **Run Application**.



6. Log in to the **Sample Reporting** application as an End User. Enter your **Username** and **Password** (Same as your Workspace credentials) and click **Sign In**.



7. You now see the Sample Reporting application home page. In your Runtime environment under **Developer Toolbar**, click **Home** to return to the home page of the APEX

development environment.

The screenshot shows the Oracle APEX Sample Reporting application. The left sidebar contains a navigation menu with items like Sample Reports, Interactive Report, Interactive Grid, Faceted Search, Cards, Classic Report, Use Cases, SQL Examples, Analytic Functions, and Administration. The main content area is titled "Sample Reporting" and describes the application's purpose: "Demonstration of reports and reporting techniques in Oracle APEX". It features a grid of eight cards, each representing a different reporting technique:

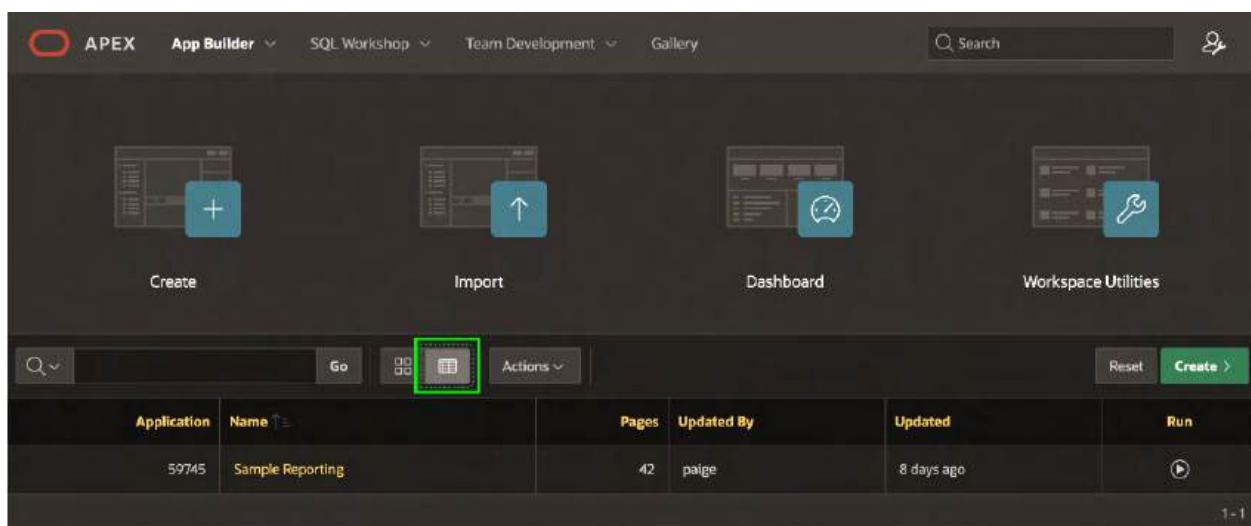
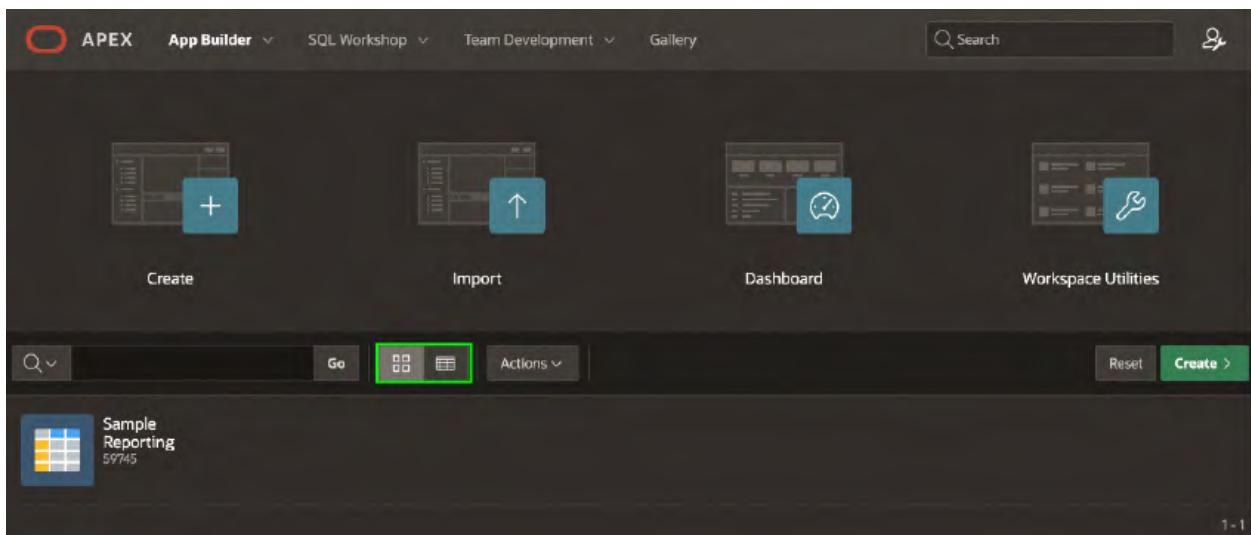
| Interactive Report | Interactive Grid | Faceted Search | Cards |
|--|--|---|---|
| Reports which enable customization by the end user | Reports which allow for in-line editing and other customization by | Report with facets to filter the result | Report displayed in cards |
| Classic Report | Use Cases | SQL Examples | Analytic Functions |
| Tabular data which can be filtered by page item values | Demonstrations of advanced APEX report techniques | Demonstrations of advanced SQL techniques | Demonstrations of various analytic functions available in Oracle SQL. |

Below the cards are social sharing icons for Twitter, LinkedIn, Facebook, and YouTube. The bottom of the screen shows standard APEX navigation and status bars.

8. Click App Builder.

The screenshot shows the Oracle APEX home page. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below the navigation is a row of icons: App Builder (highlighted with a red box), SQL Workshop, Team Development, and Gallery. The bottom section displays "Top Apps" (Sample Reporting), "Top Users" (Roopesh Thokala), and a summary of "1 Applications", "3 Tables", and "1 Developers".

9. Toggle between the **View Icons** and the **View Reports** buttons. You see that applications are displayed along with their icons and report format.

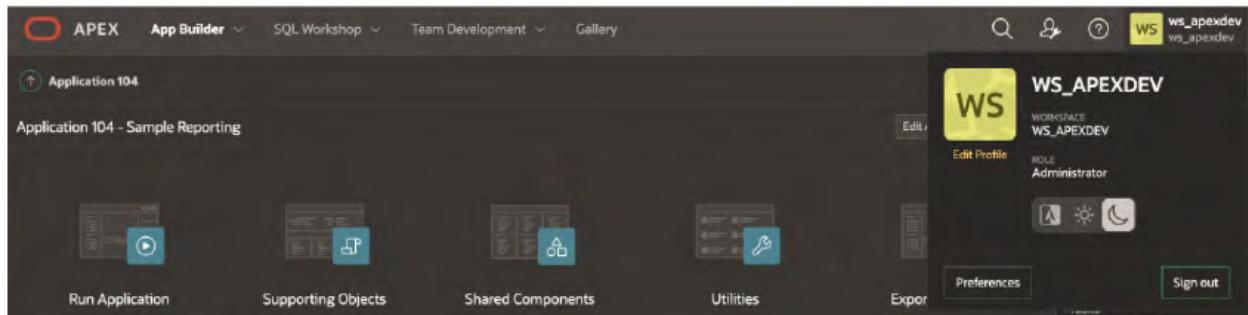


Configure the Target System

In this Lab, You will enable the **Target System** to **ORDS**, and then you will copy the URL and paste it into your notepad.

Note: You need to provision a new **APEX Service** or **APEX on Autonomous Database** and create a new workspace in order to perform this lab.

1. Log in to your **Target System**.



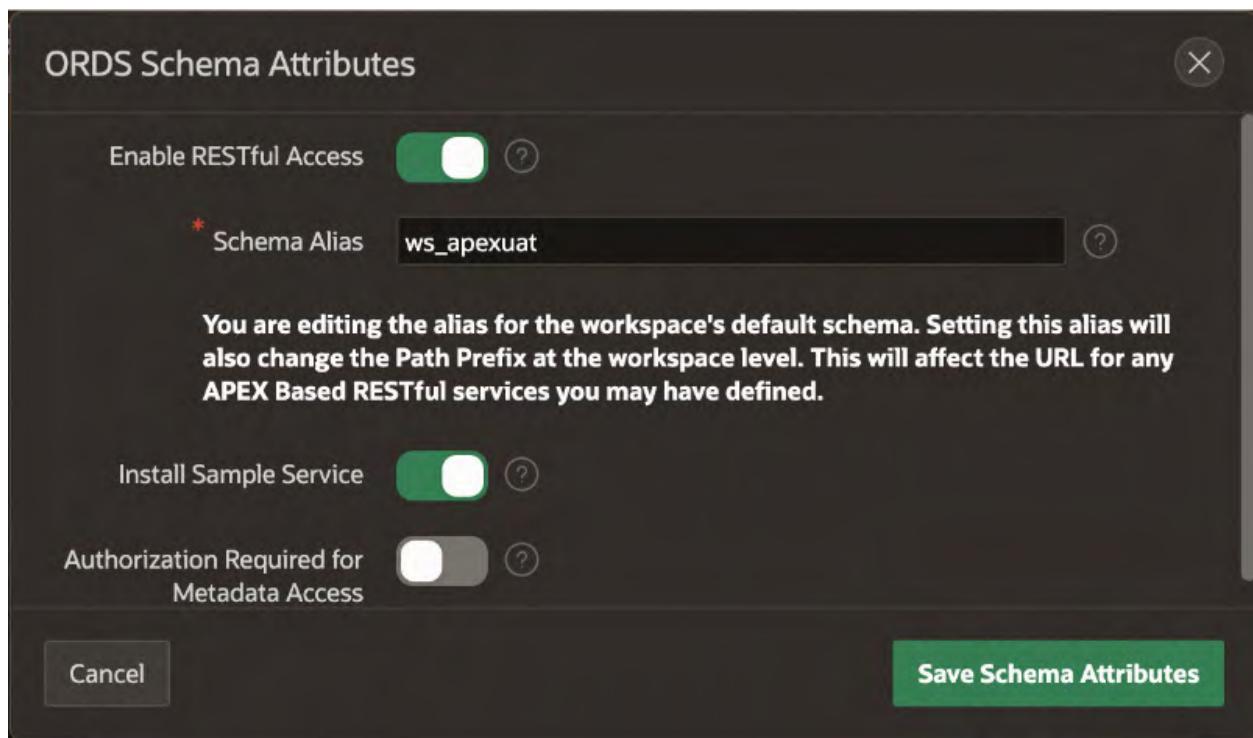
2. Navigate to **SQL Workshop > RESTful Services**.



3. Click **Register Schema with ORDS**.



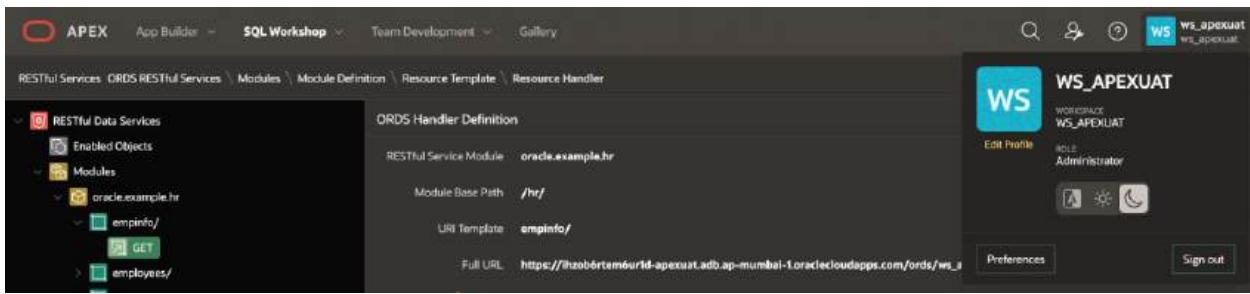
4. Review the **ORDS Schema Attributes** and click **Save Schema Attributes**. Now, you see a message that the schema has been successfully registered.



5. Navigate to **SQL Workshop > Restful Services**. Expand **RESTful Data Services > Modules > oracle.example.hr > empinfo/ GET**. Copy the **Full URL** and paste it into your **Clipboard or Notepad**.

The screenshot shows the Oracle SQL Workshop interface with the 'ORDS RESTful Services' section selected. On the left, a tree view shows 'Enabled Objects', 'Modules' (expanded to show 'oracle.example.hr' which has 'empinfo/' selected), and 'Privileges/Roles'. On the right, the 'ORDS Handler Definition' panel shows the 'RESTful Service Module' as 'oracle.example.hr', 'Module Base Path' as '/hr/', 'URI Template' as 'empinfo/', and the 'Full URL' as 'https://.../hr/empinfo/'. The 'Method' is set to 'GET', 'Source Type' to 'Query', 'Format' to 'CSV', and 'Pagination Size' to '25'. A 'Comments' text area is present. At the top right, there are buttons for 'Schema' (set to 'WS_APEXUAT'), 'Cancel', 'Delete', and 'Apply Changes' (green).

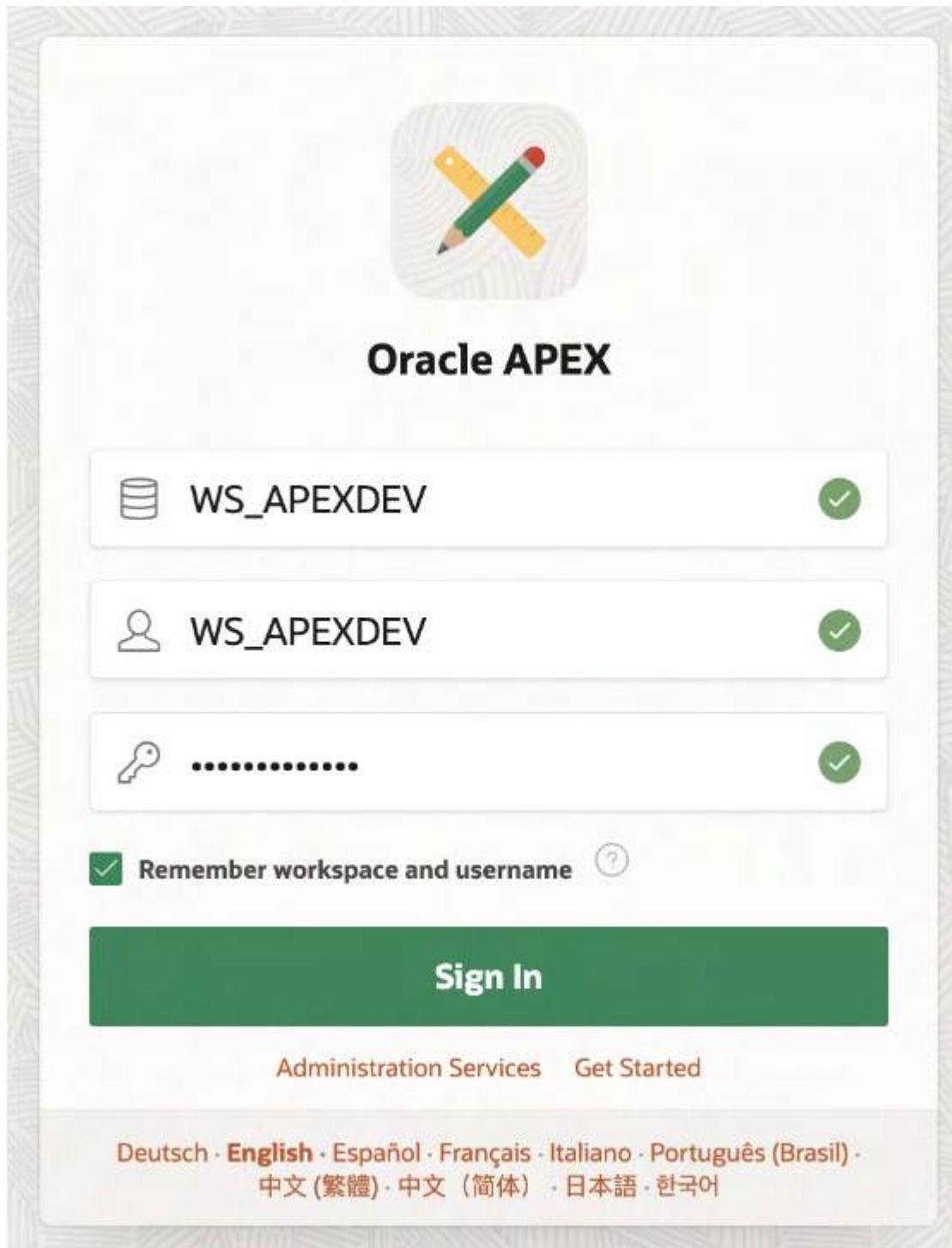
6. Click **Sign Out**.



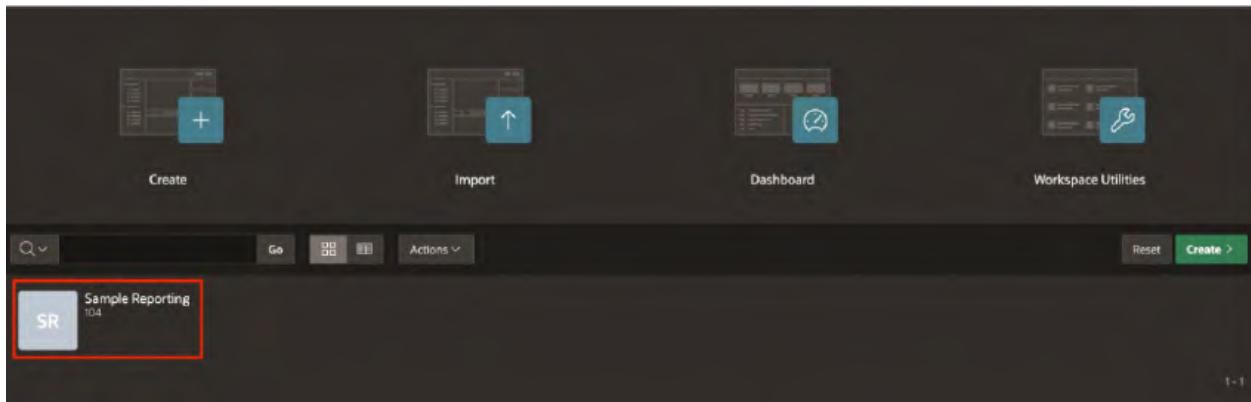
Remote Deployment

Your application is now ready for deployment. Perform the following steps:

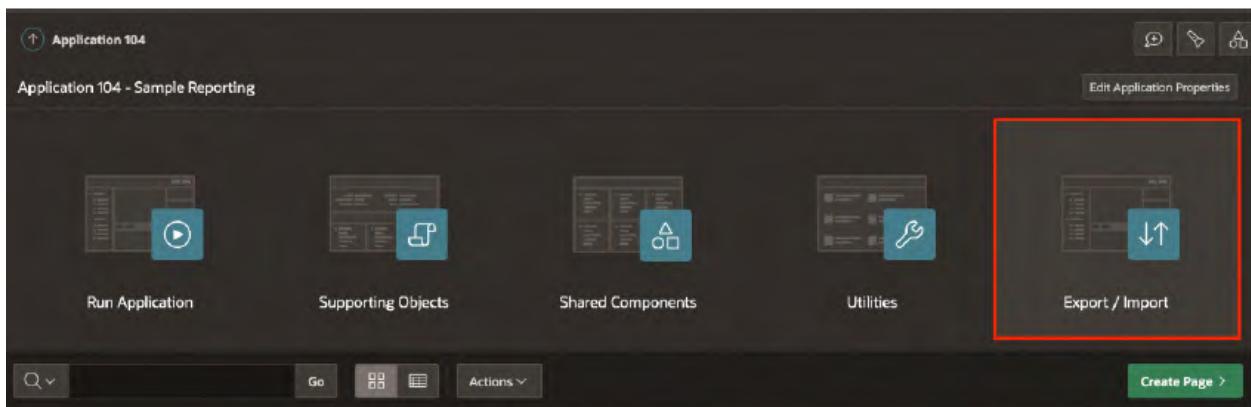
1. Log in to your development environment (WS_APEXDEV in this example)



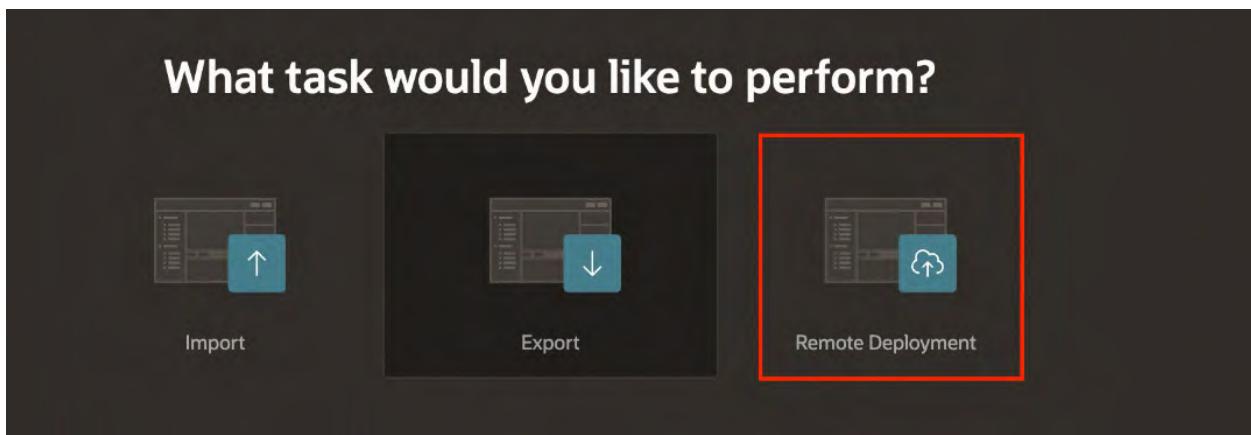
2. Navigate to your application home page. select **App Builder** and then click **Sample Reporting**.



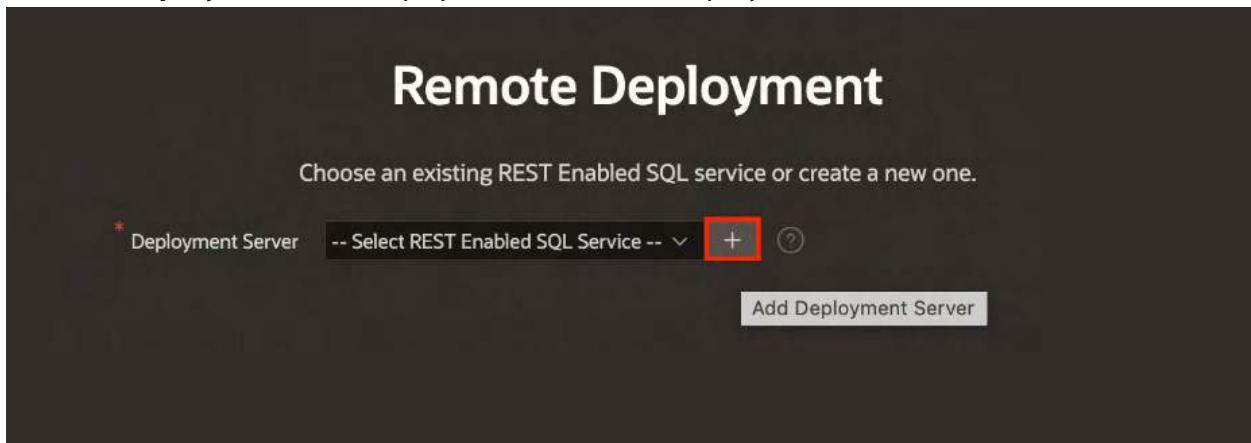
3. Click **Export / Import**.



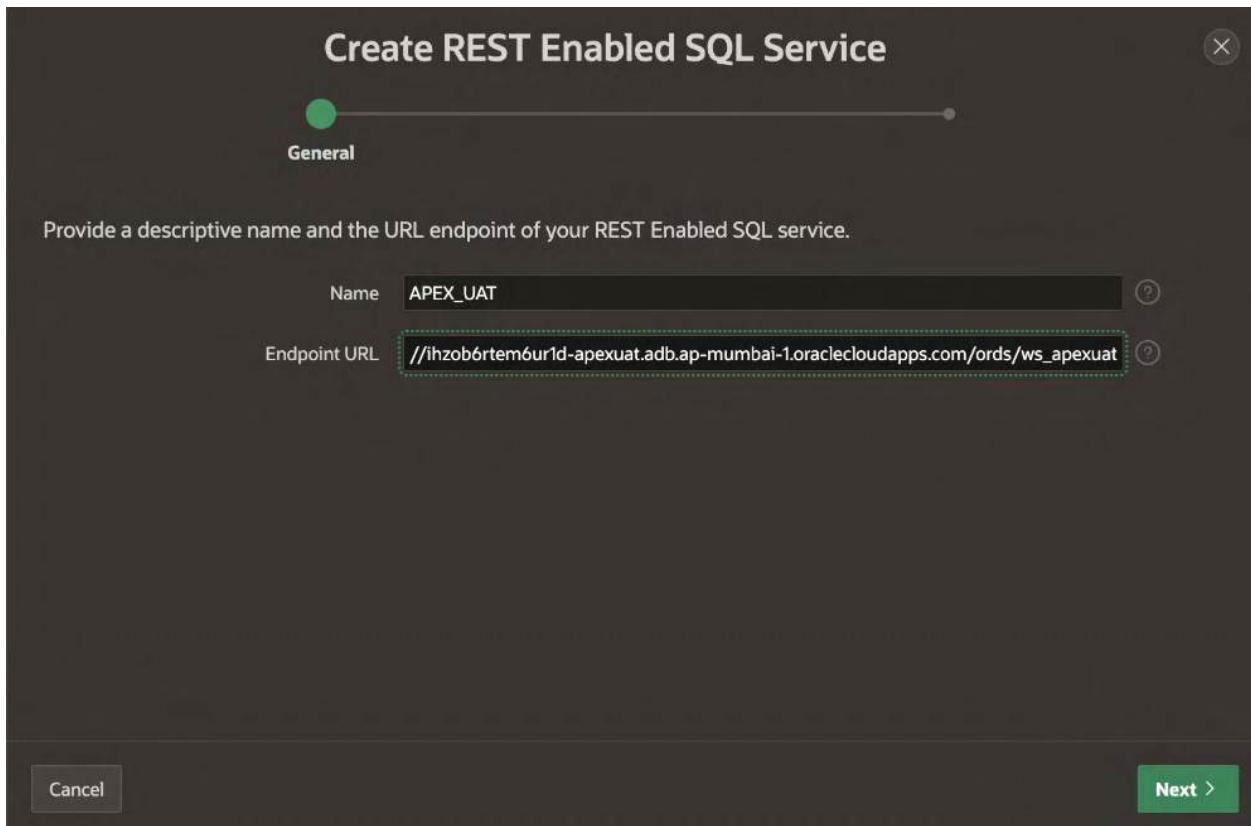
4. Select **Remote Deployment** and click **Next**.



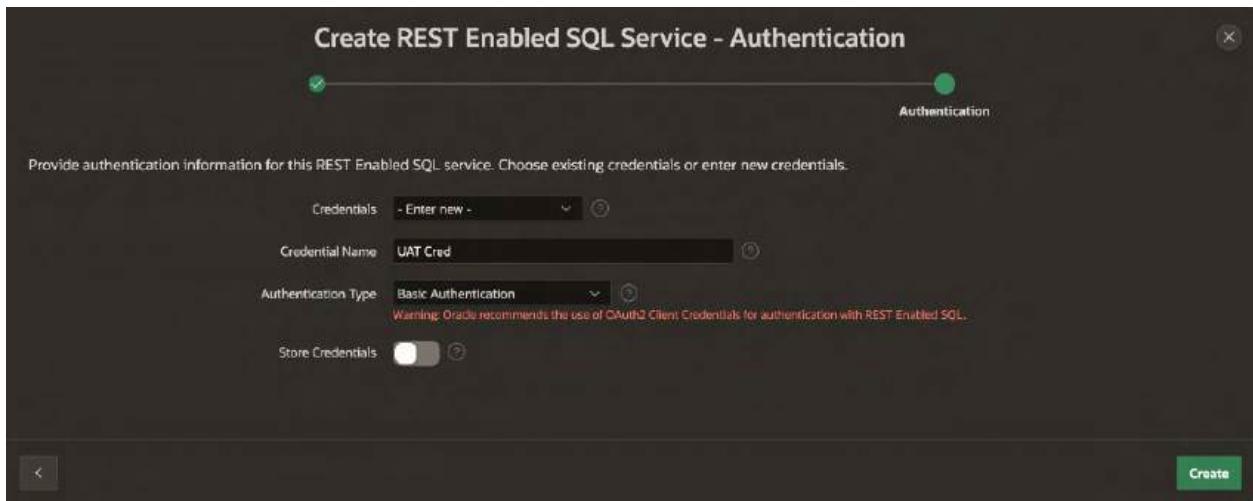
5. Choose an existing **REST Enabled SQL Service** or create a new one. In this example, click the **Add Deployment Server** (+) icon next to the Deployment Server select list.



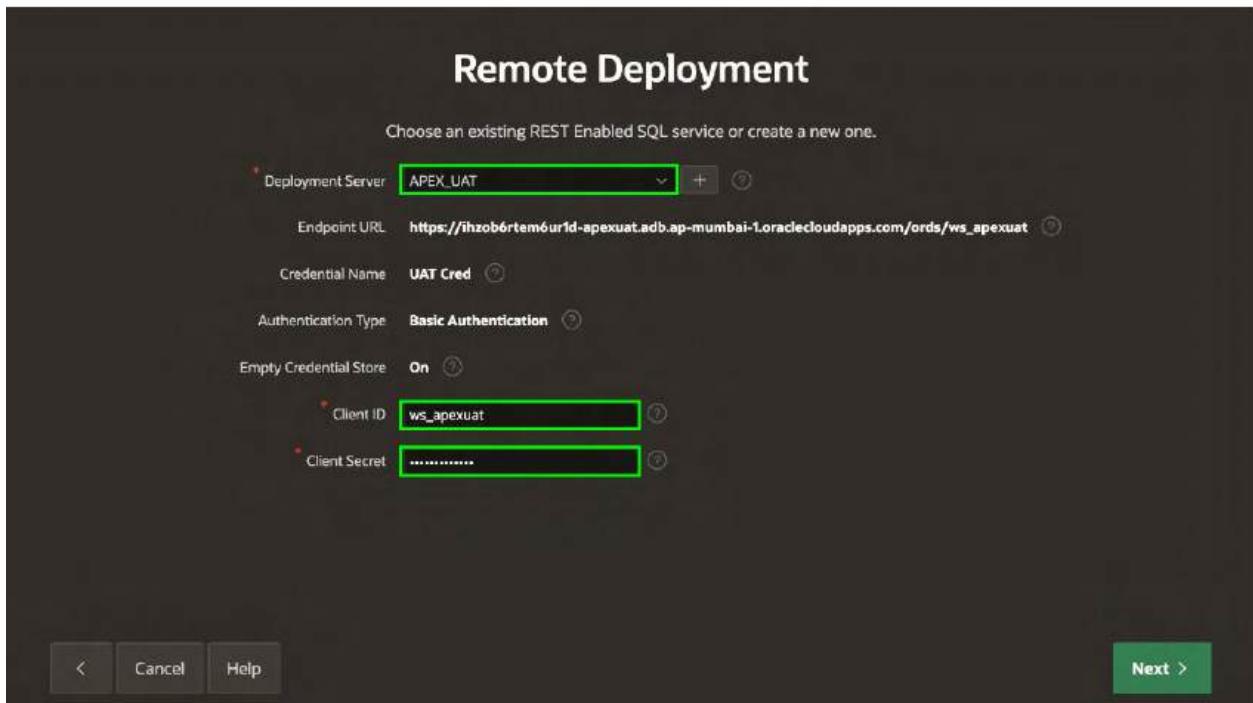
6. Enter the base URL for the **REST Enabled SQL Service** defined for your deployment server (WS_APEXUAT in this example). The schema in your deployment server has already been enabled for use with ORDS RESTful Data Services in prerequisites. For **Endpoint URL**, paste the base URL copied from the deployment instance in the previous step(**Task 4**). The base URL should include the ORDS context root and schema URL prefix. For example, <https://host:8096/ords/apexstage>. Then, click Next.



- Enter **UAT Cred** for Credential Name, select **Basic Authentication** for Authentication Type, and turn off the **Store Credentials** switch. Click **Create**.



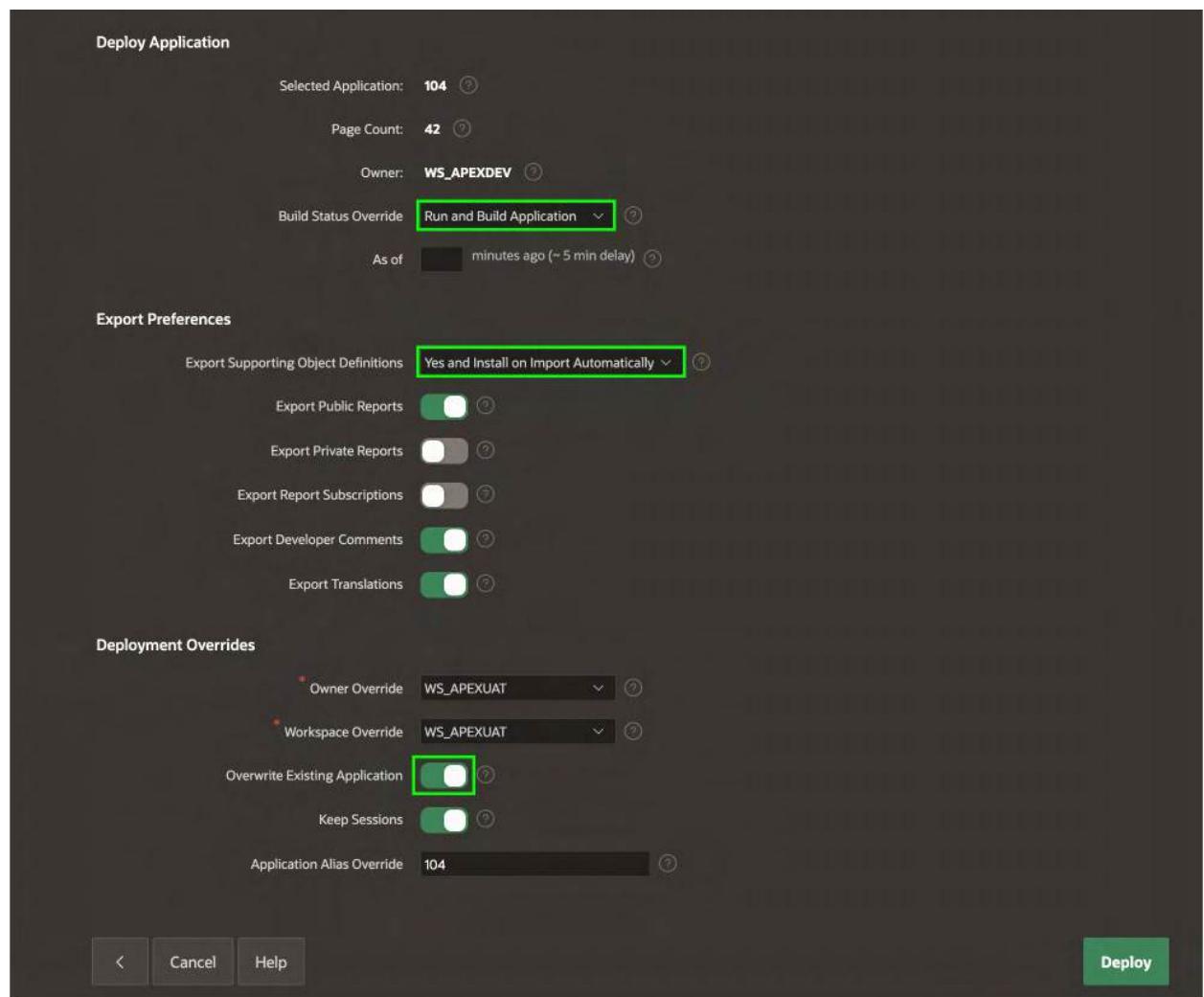
- Now you see a message that REST Enabled SQL Service has been created. From the Deployment Server select list, choose the one that you just created. Enter the values for **Client ID** and **Client Secret** and click **Next**. In this example, you use **Basic Authentication**. Therefore, enter the **Database user name** (deployment schema) for Client ID and the corresponding password for Client Secret. Then, Click **Next**.



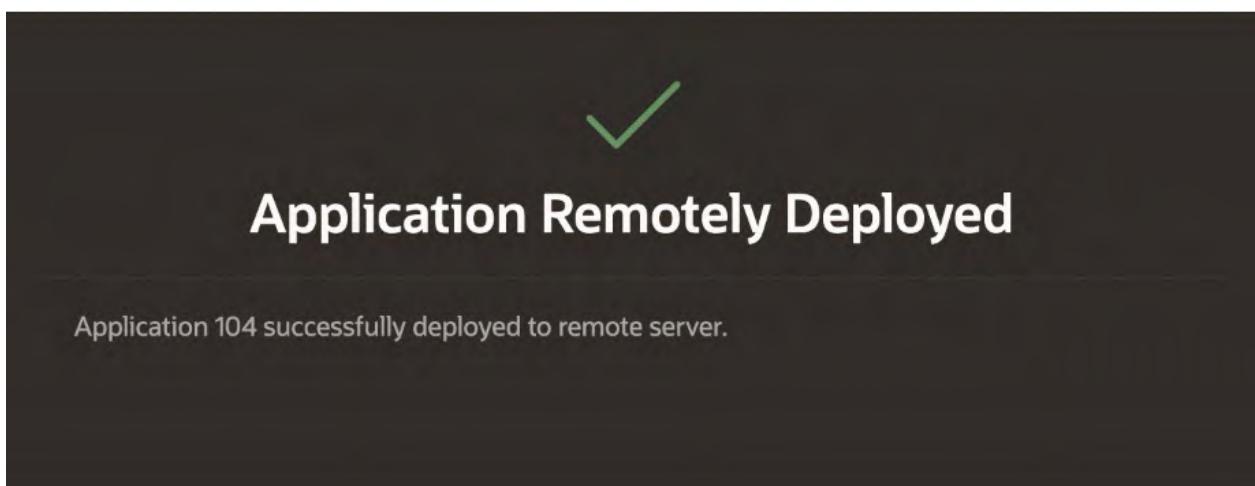
- The Remote Deployment Options page appears.

- Under **Deploy Application**: Set **Build Status Override** to **Run and Build Application**.
- Under **Export Preferences**: For **Export Supporting Object Definitions**, select **Yes and Install on Import Automatically**.
- Under **Deployment Overrides**: In this example, even though the application does not yet exist on the target system, turn the **Overwrite Existing Application option On** so that you can preserve the application ID, and enable this application for future overwrites.

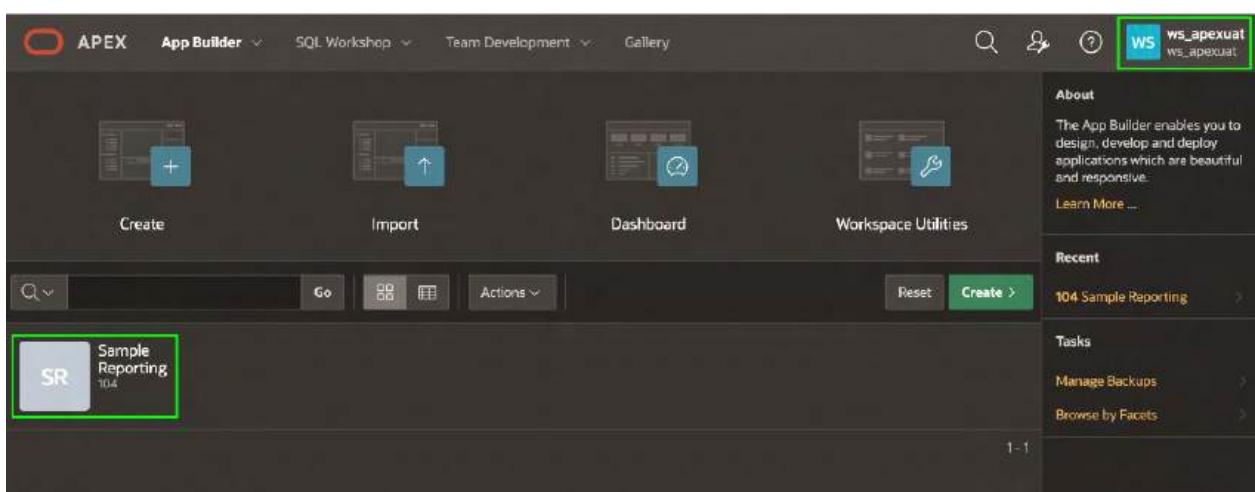
After making the above selections, click **Deploy**.



10. You see the Remote Deployment Successful message. Your application has been successfully deployed to the remote server.



11. Log in to your remote APEX instance. Make sure that the application is available and with the same application id as in the source system.



Summary

You now know how to deploy your Application from Development to UAT.