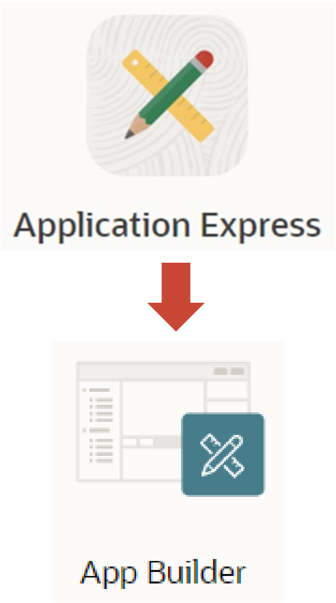
Creating a Simple Autonomous Database Application in Oracle APEX



Tutorial: Creating a Simple Database Application in Oracle APEX

Topic	Details
Overview	In this tutorial, you will create a simple database application in Oracle APEX using App Builder.
Key Concepts	• Create an application using the Application Wizard • Create additional pages using the Page Wizard • Use existing sequences to populate primary key columns • Create Lists of Values to populate foreign key columns. • Create Master Detail Forms • Upload an image file to display on a page in the application
Difficulty	Intermediate
Duration	Approximately 90 minutes
	Note: Learners will require access to an active APEX account with tables and data uploaded from the SQL schema script file (which can be accessed from Course Resources of either database course). This tutorial was built using APEX 22.2

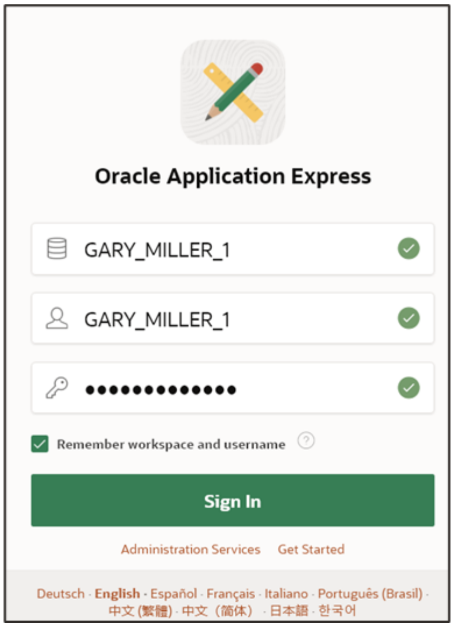
Part 1: Introduction and Getting Started

This tutorial will demonstrate how to create a database application using the App Builder component of Oracle Application Express (APEX) on a Cloud Autonomous Database.

The application created in this tutorial will allow end-users to add, modify and delete data in the existing Jobs, Employees and Departments tables without the need to run SQL queries.

Step 1: Log on to APEX

1. Log on to APEX using your Autonomous Database APEX credentials.



Oracle Application Express

GARY_MILLER_1

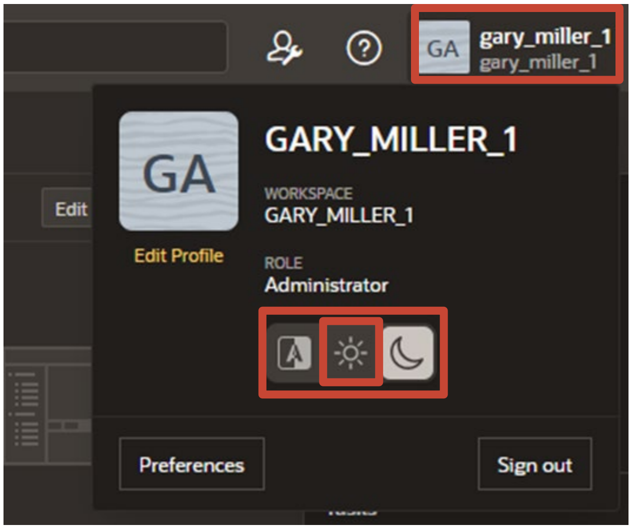
GARY_MILLER_1

••••••••••••••

☑ Remember workspace and username

Sign In
Administration Services
Get Started
Deutsch · English · Español · Français · Italiano · Português (Brasil) ·
中文 (繁體) · 中文（简体） · 日本語 · 한국어

2. By default, APEX is in “Dark Mode”. The screenshots in this lab are displayed in “Light Mode”. You can select your chosen mode by clicking your workspace name from the top right of APEX home page and clicking your choice.



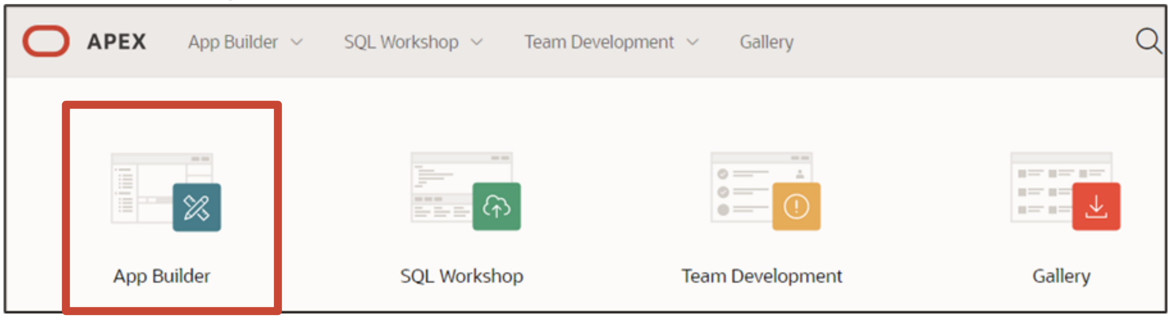
GARY_MILLER_1

WORKSPACE
GARY_MILLER_1

ROLE
Administrator

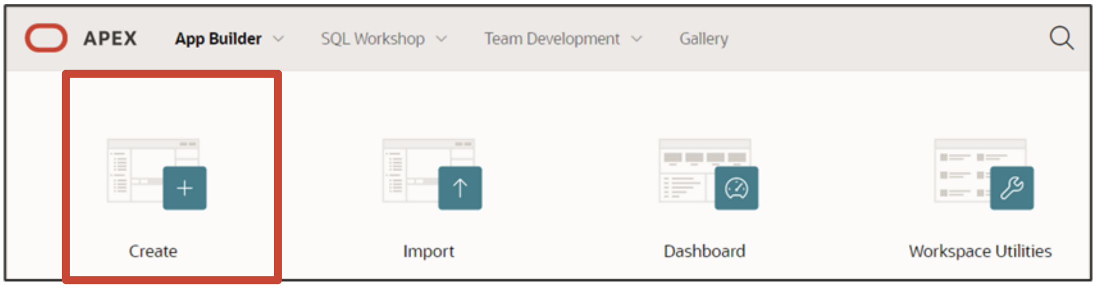
Step 2: Accessing Application Builder

1. In the home page of APEX, click the **App Builder** icon.

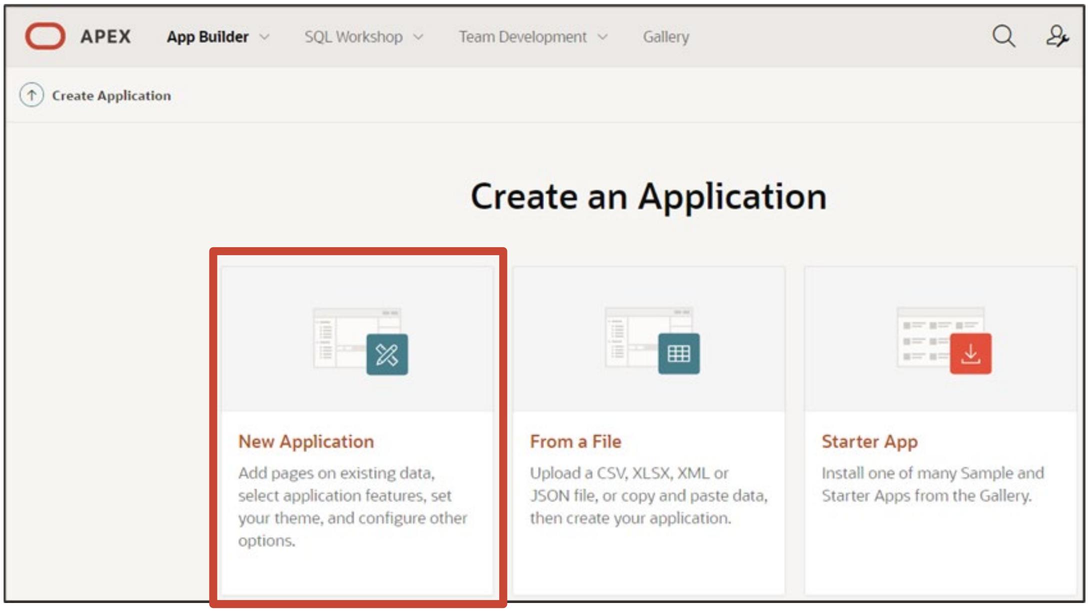


Step 3: Create a New Application

1. Click the **Create** icon to create a new application.



2. Click **New Application**.



Create an Application

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

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

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

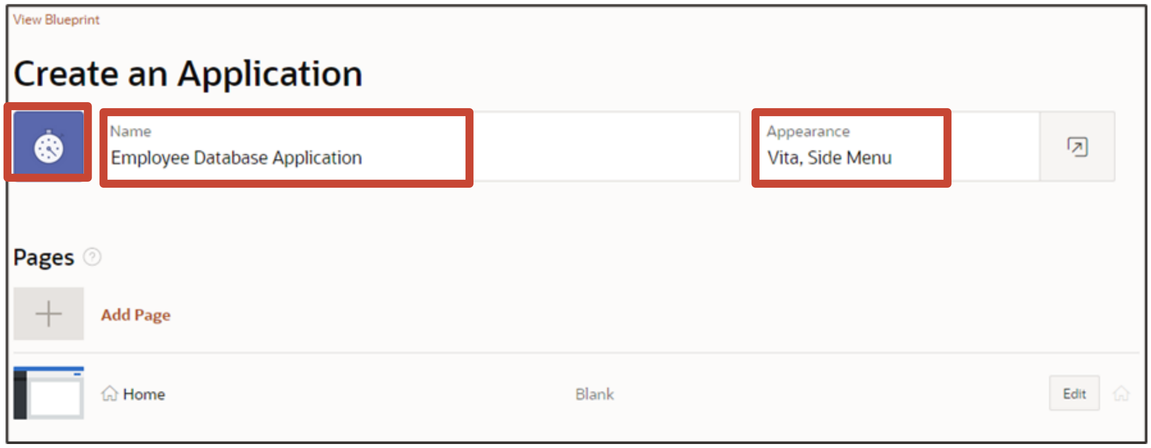
Step 4: Name the Application

Follow the steps below to continue creating the application:

1. **Name:** Employee Database Application

   Appearance: Accept default Vita theme

   Click the "**Set Icon**" button.



View Blueprint

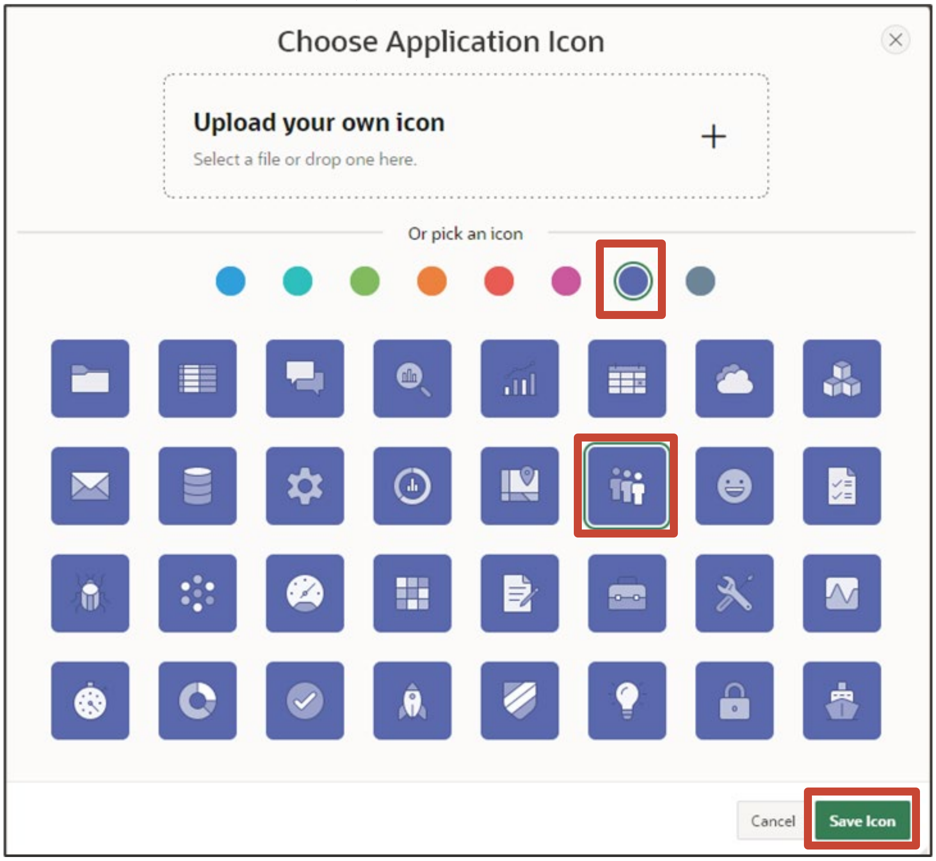
Create an Application

Name: Employee Database Application

Appearance: Vita, Side Menu

Pages

2. Select an icon and its color to use for your application, then click **Set Application** Icon. (You can also upload an image to use as the app icon by dragging an image file from your local machine to the "Upload your own icon" area.) Click **Save Icon**.



Choose Application Icon

Upload your own icon

Select a file or drop one here.

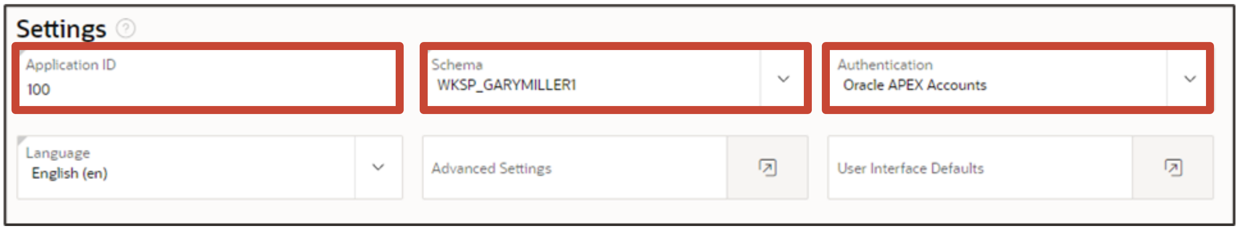
Or pick an icon

3. Scroll down to confirm the **Settings** options:

   Application ID: Do not change the application number. APEX creates an application number by default, and it is recommended that you do not change this number.

   Schema: your schema workspace will be displayed by default.

   Authentication: Oracle APEX Accounts



Settings

Application ID: 100

Schema: WKSP_GARYMILLER1

Authentication: Oracle APEX Accounts

Language: English (en)

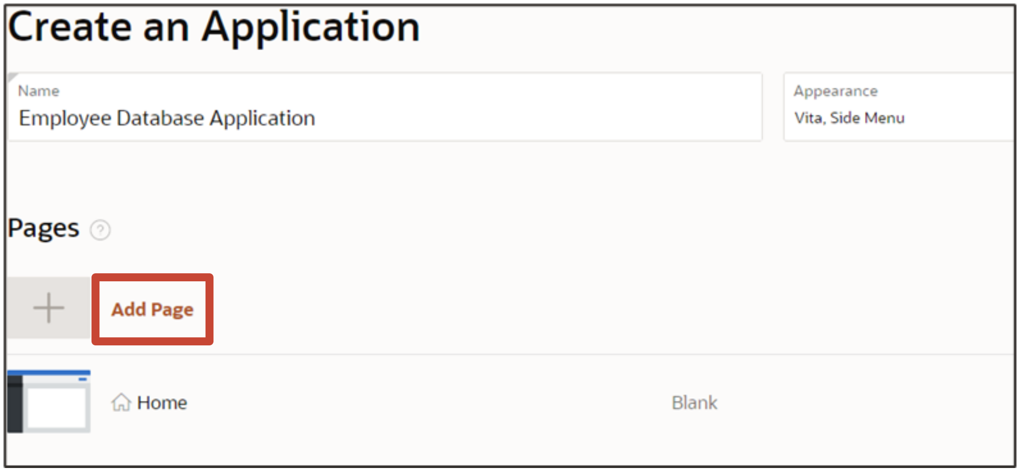
   Note: Do **NOT** click Create Application at this time.

Step 5: Add a Page to the Application

A “Home” page is added automatically to your application, this will be the parent page for any other page that we add.

In this step, we will add a page that will display the data in the jobs table.

1. Scroll up to the **Add Page** link and click it.



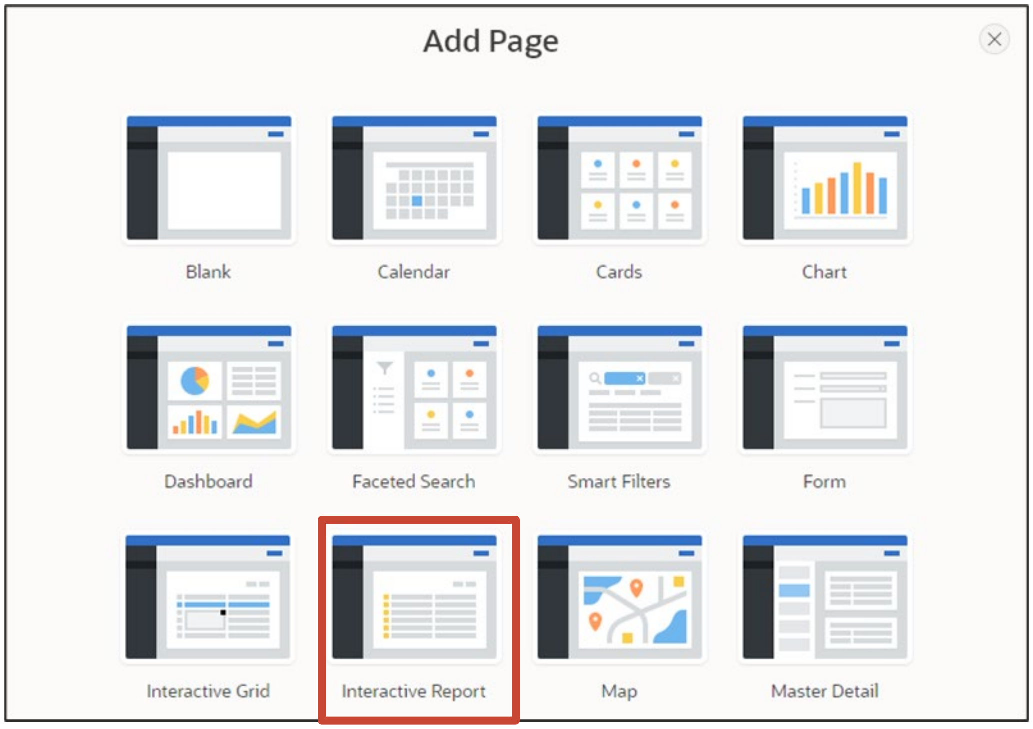
Create an Application

Name: Employee Database Application

Appearance: Vita, Side Menu

Pages

2. Select **Interactive Report**.



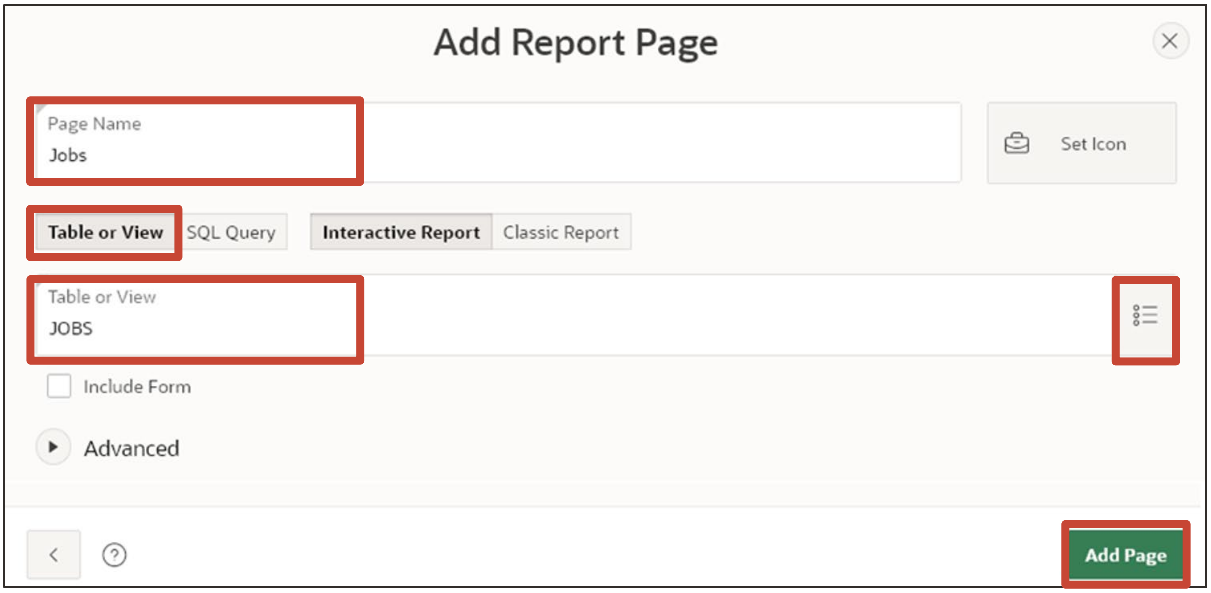
3. In the **Add Report Page**, enter the following:

   Page Name: Jobs

   Table or View: Click the list icon, find the Jobs table, and click it.

   Click **Set Icon** and select the icon you want to use for the page. (The example below uses the fa-briefcase icon).

   Click **Add Page**.

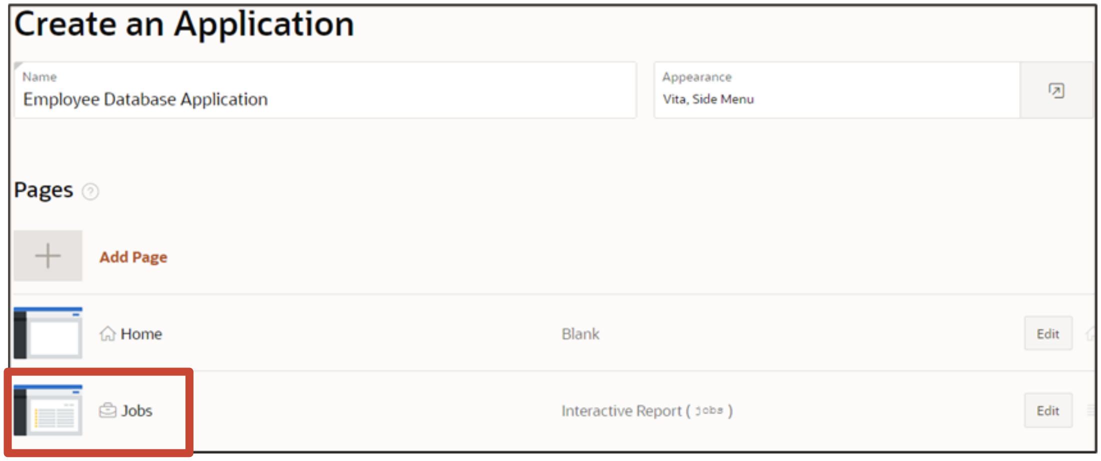


Page Name: Jobs

Table or View

Table or View: JOBS

4. The Jobs page should show in your application.



Create an Application

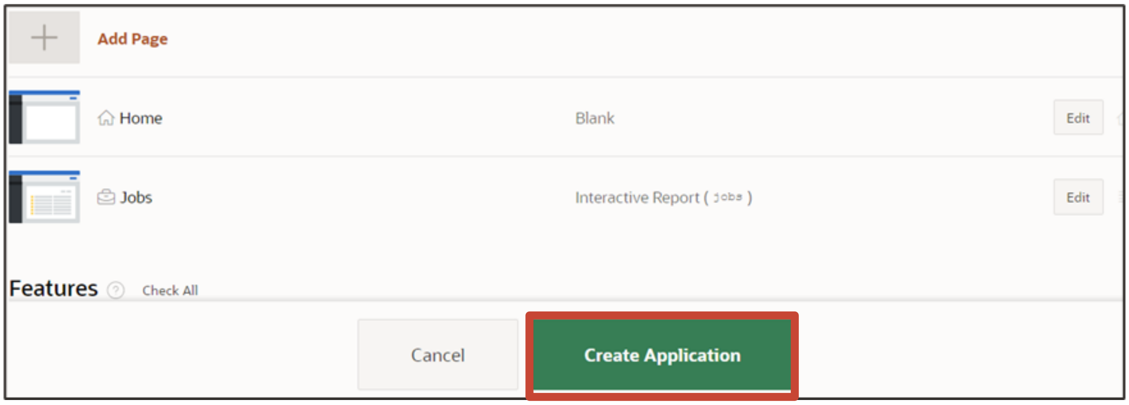
Name: Employee Database Application

Appearance: Vita, Side Menu

Pages

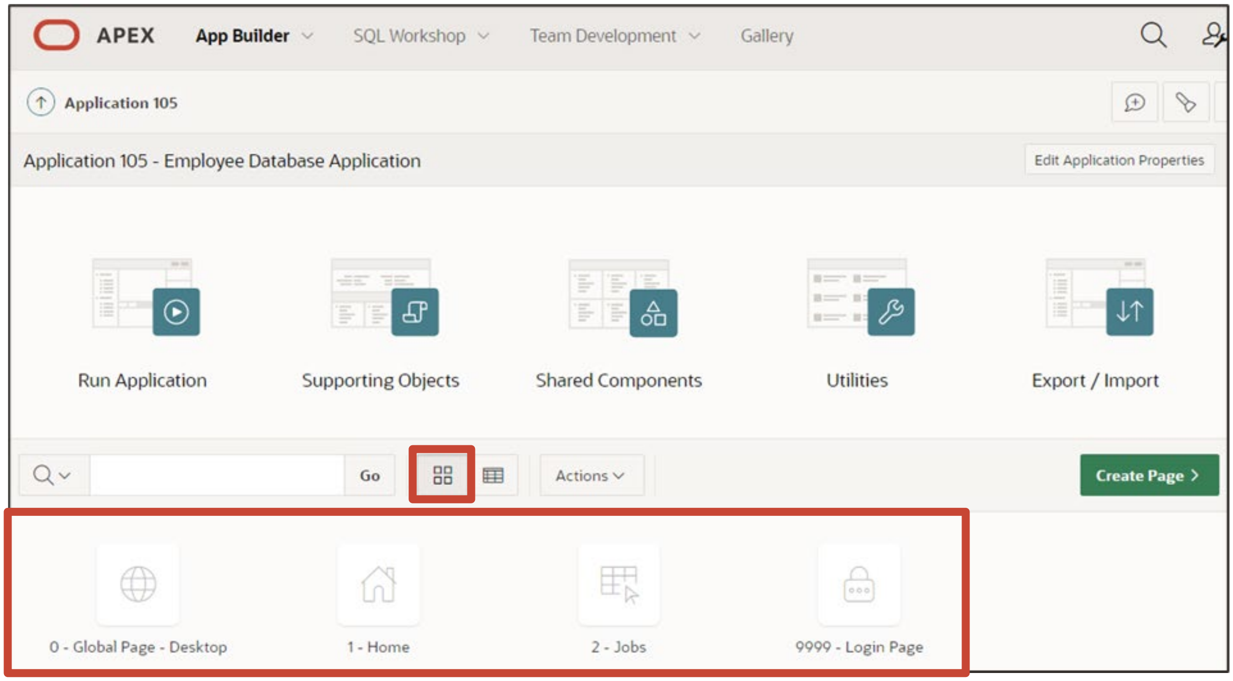
Home	Blank	Edit
Jobs	Interactive Report (Jobs)	Edit

5. Click the **Create Application** button.



Home	Blank	Edit
Jobs	Interactive Report (Jobs)	Edit

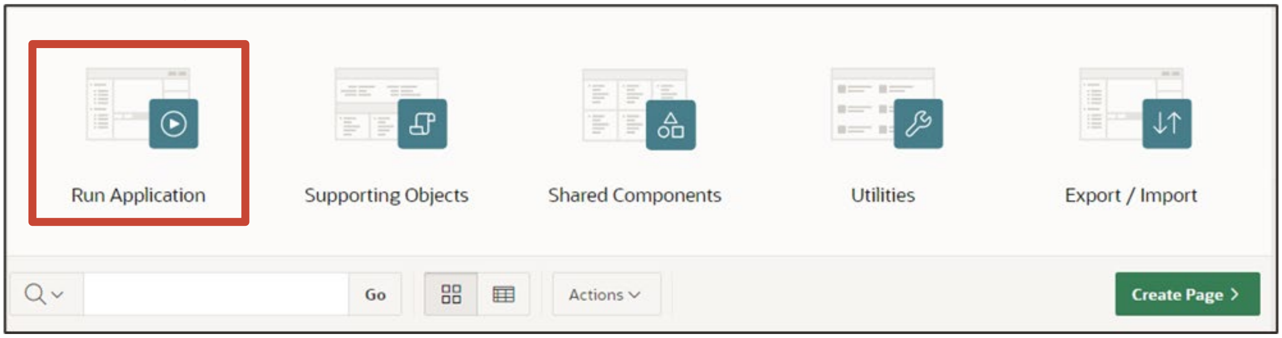
6. You will see your application and the pages it contains. If your pages are displayed as a list, click the **view icons** button. (**Note**: The “Application” number may differ)

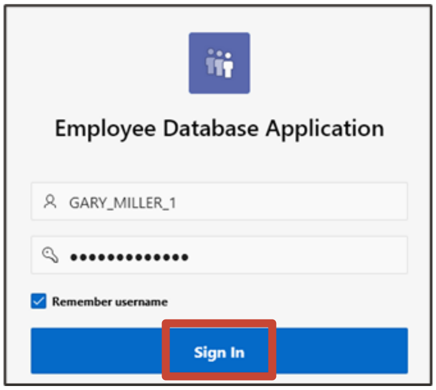


Application 105 - Employee Database Application

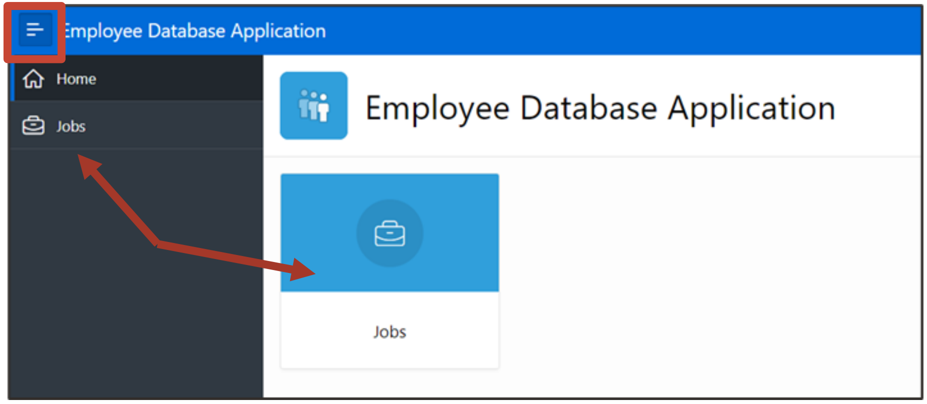
Step 6: Run and Test Application

1. Click the **Run Application** icon. Your application will open in a new browser window (or tab). Login using your APEX username and password.

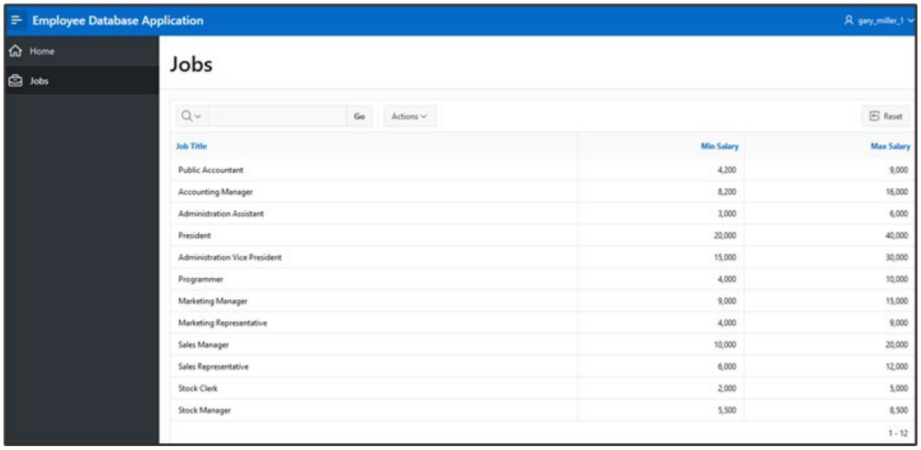




2. Once the app is displayed in the browser, click the **Navigation Menu** icon to expand the navigation pane, then select the **Jobs** page from the navigation pane on the left or by clicking the page icon.



3. The data in the jobs table will be displayed.



Jobs

Job Title	Min Salary	Max Salary
Public Accountant	4,200	9,000
Accounting Manager	8,200	16,000
Administration Assistant	3,000	6,000
President	20,000	40,000
Administration Vice President	15,000	30,000
Programmer	4,000	10,000
Marketing Manager	9,000	15,000
Marketing Representative	4,000	9,000
Sales Manager	10,000	20,000
Sales Representative	6,000	12,000
Stock Clerk	2,000	5,000
Stock Manager	5,500	8,500

1 - 12

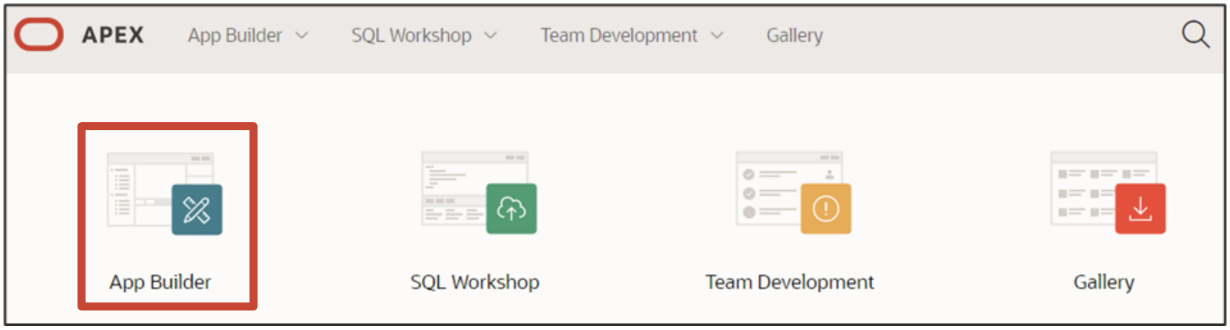
4. Close the browser tab (or window) showing the running application.

Part 2: Add an Employees Page Using the Create Page Wizard

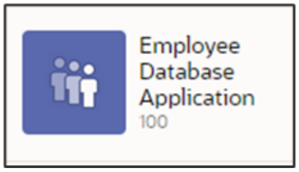
Step 1: Edit the Application and Start the Page Wizard

In APEX, you can access your application at any time by clicking the App Builder tab or icon to return to Application Builder home page.

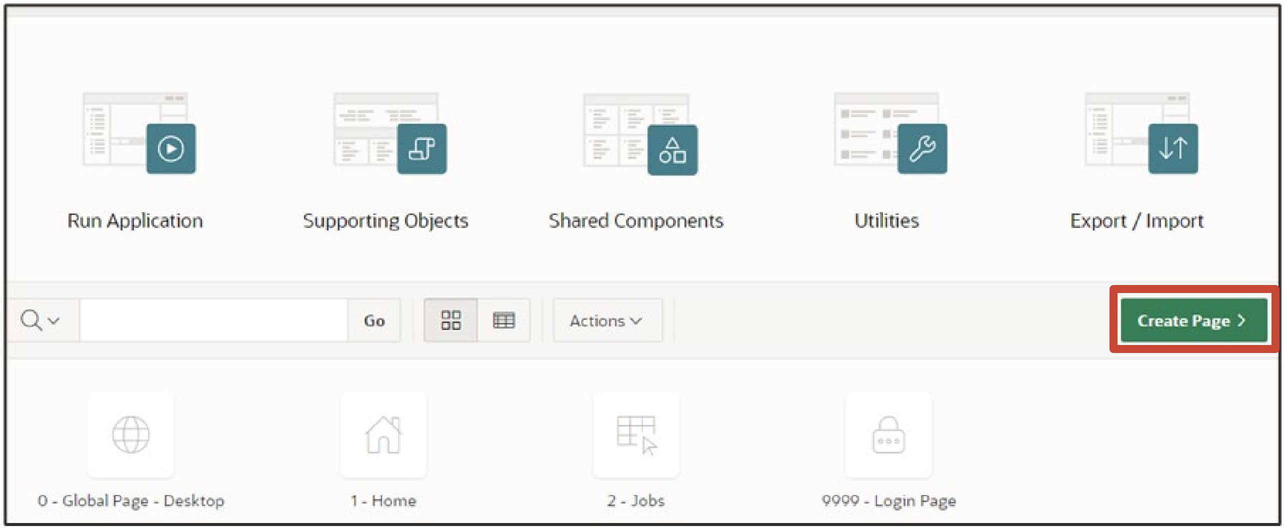
1. If your application home page is not displayed, click the **App Builder** icon.



2. Click the **Employee Database Application** Icon.

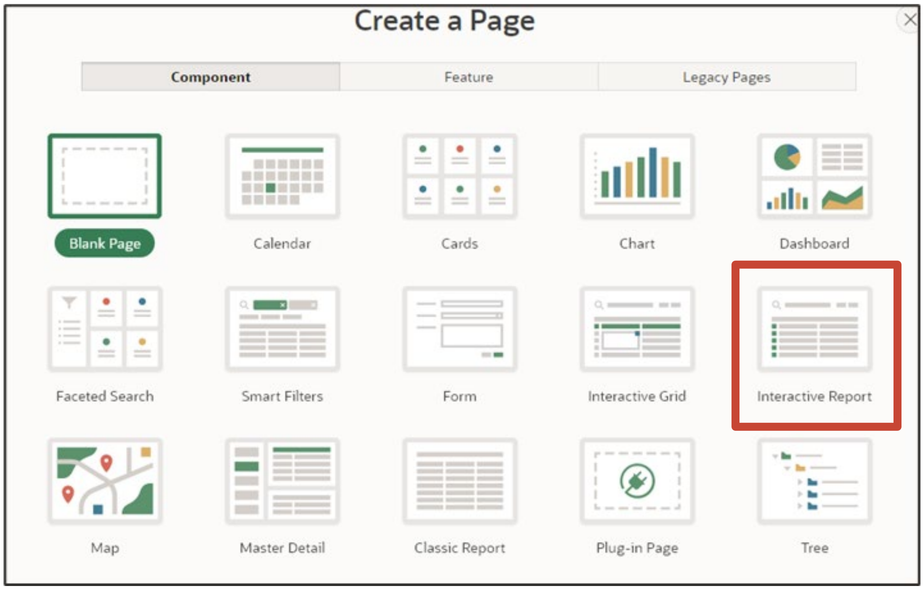


3. From the application home page, click the **Create Page >** button to start the Page Wizard.



Step 2: Edit Select Page Type

1. Click **Interactive Report**.



2. Enter the following properties for the Page Definition:

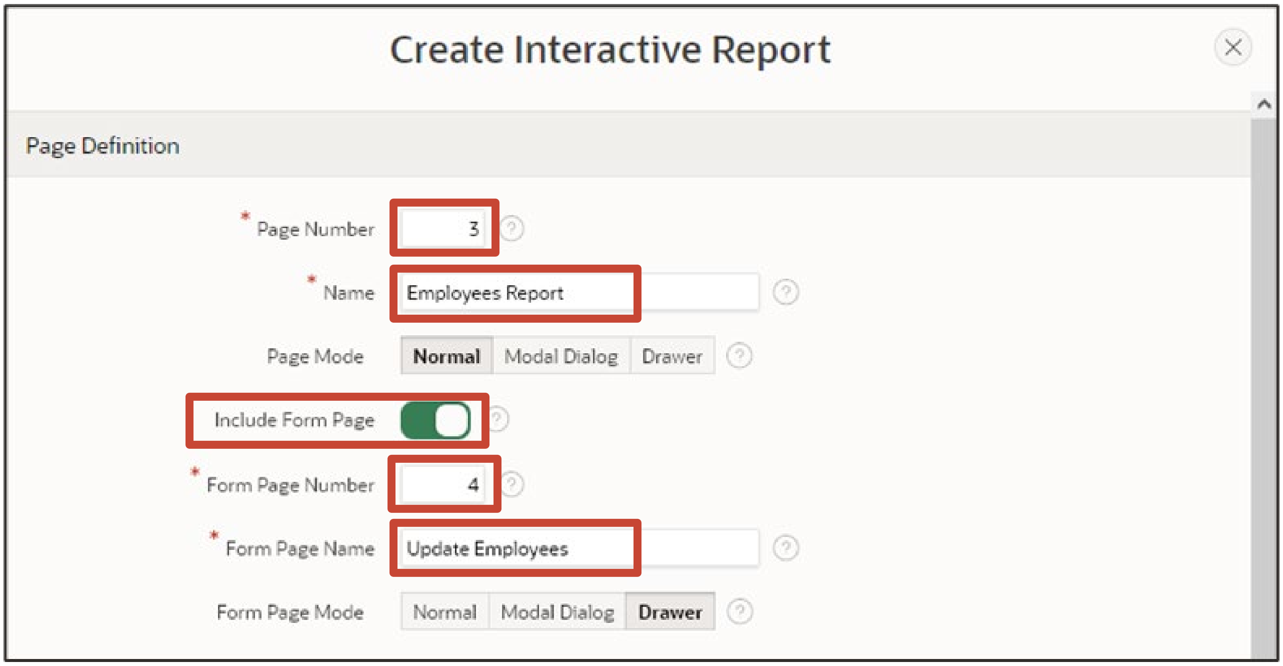
   Page Number: 3

   Page Name: Employees Report

   Include Form Page: Move the slider to make it green (on position)

   Form Page Number: 4

   Form Page Name: Update Employees



Create Interactive Report

Page Definition

* Page Number	3
* Name	Employees Report
Page Mode	**Normal** | Modal Dialog | Drawer
Include Form Page	On
* Form Page Number	4
* Form Page Name	Update Employees
Form Page Mode	Normal | Modal Dialog | **Drawer**

Step 3: Select Data Source

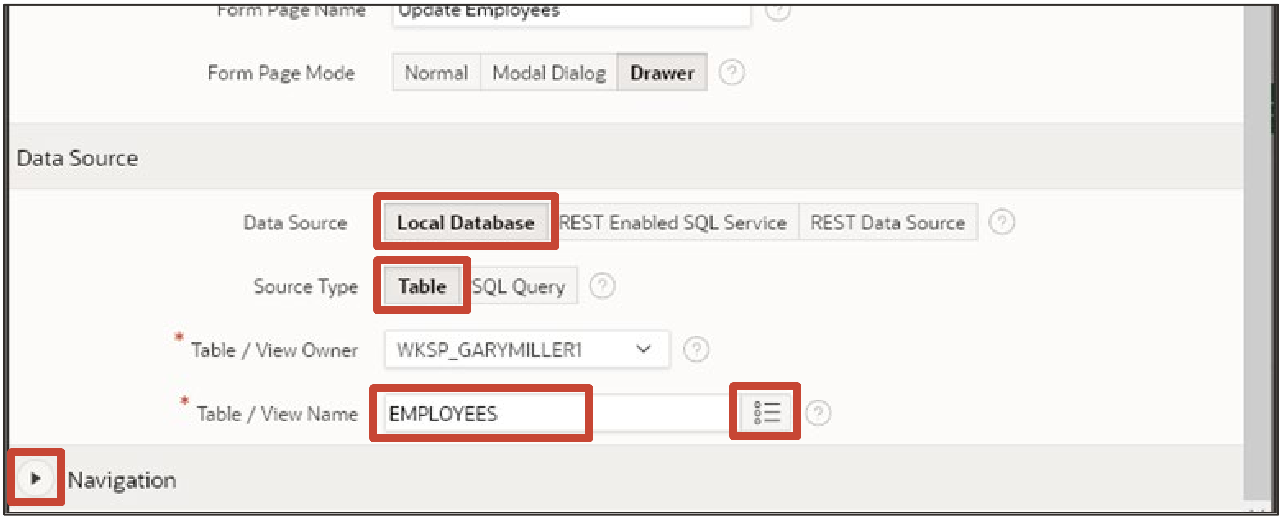
1. Scroll down to the **Data Source** properties and enter the following properties:

   Data Source: Local Database

   Source Type: Table

   Table / View Name: Click the list icon and select the **EMPLOYEES** table

   Then expand the **Navigation** properties.



Form Page Name: Update Employees

Form 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: WKSP_GARYMILLER1

* Table / View Name: EMPLOYEES

Navigation

Step 4: Add Navigation

1. Scroll down to the Navigation properties and check the following properties are selected:

   Use Breadcrumb: Move the slider to make it green (on position)

   Breadcrumb Parent Entry: empty

   Use Navigation: Move the slider to make it green (on position)

   Navigation Preference: “Create a new entry”

   For Parent Navigation Menu Entry: - No Parent Selected -

   You can select a page icon by clicking the **Select Icon** button

Click the **Next** button.

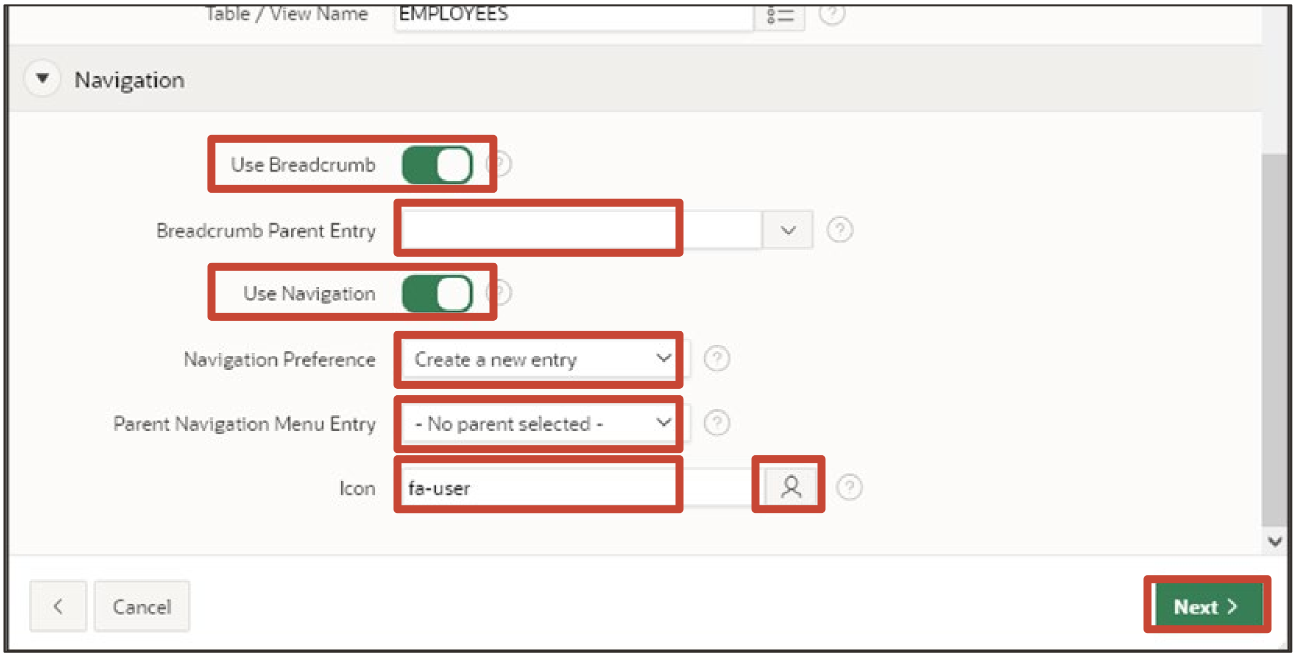


Table / View Name: EMPLOYEES

Navigation

Use Breadcrumb: On

Breadcrumb Parent Entry:

Use Navigation: On

Navigation Preference: Create a new entry

Parent Navigation Menu Entry: - No parent selected -

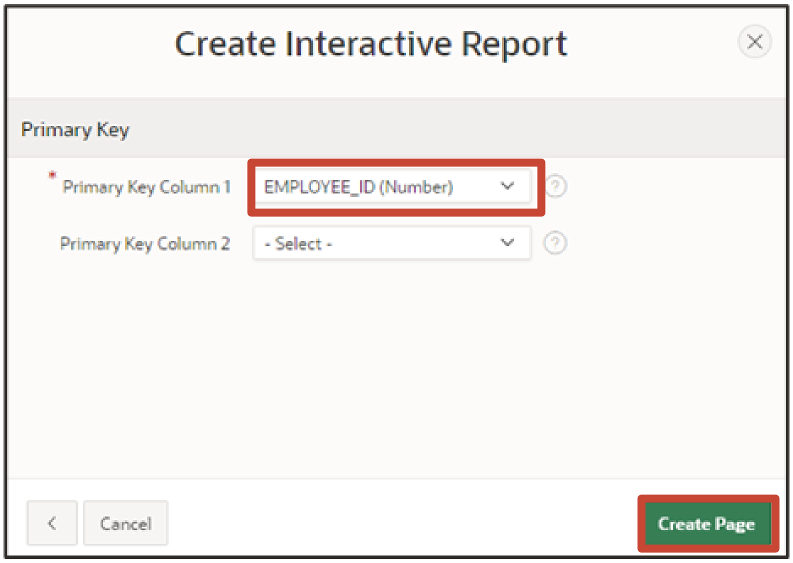
Icon: fa-user

Step 5: Select Primary Key

1. Check the following properties:

   Primary Key Column 1: EMPLOYEE_ID (Number)

   Click **Create Page**.



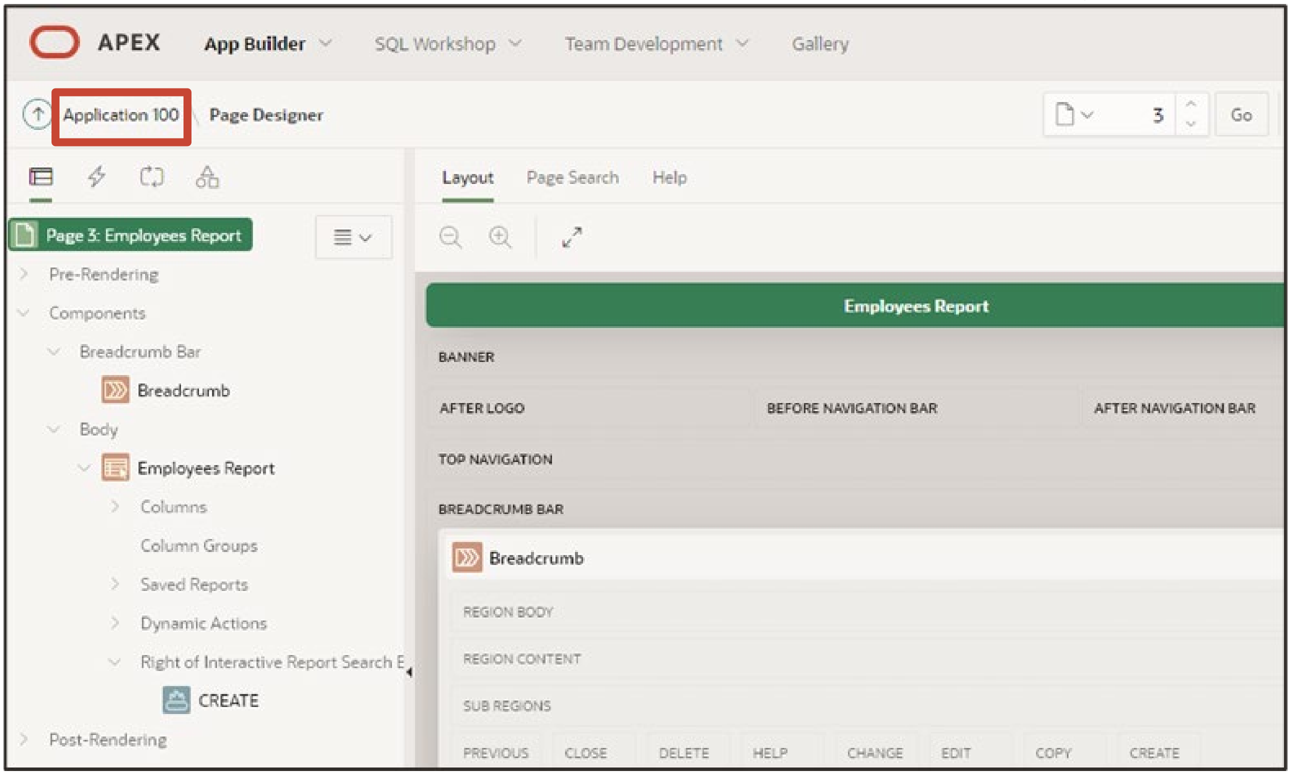
Create Interactive Report

Primary Key

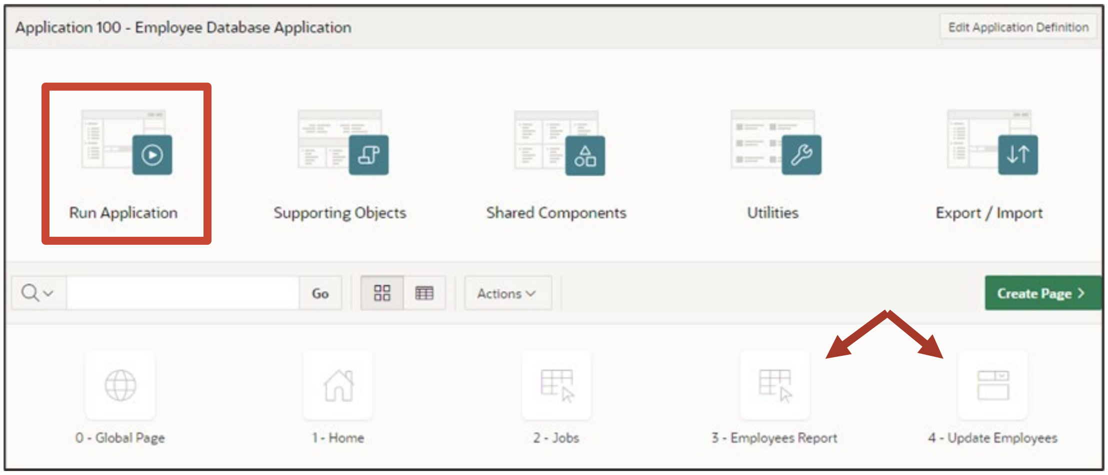
* Primary Key Column 1: EMPLOYEE_ID (Number)

Primary Key Column 2: - Select -

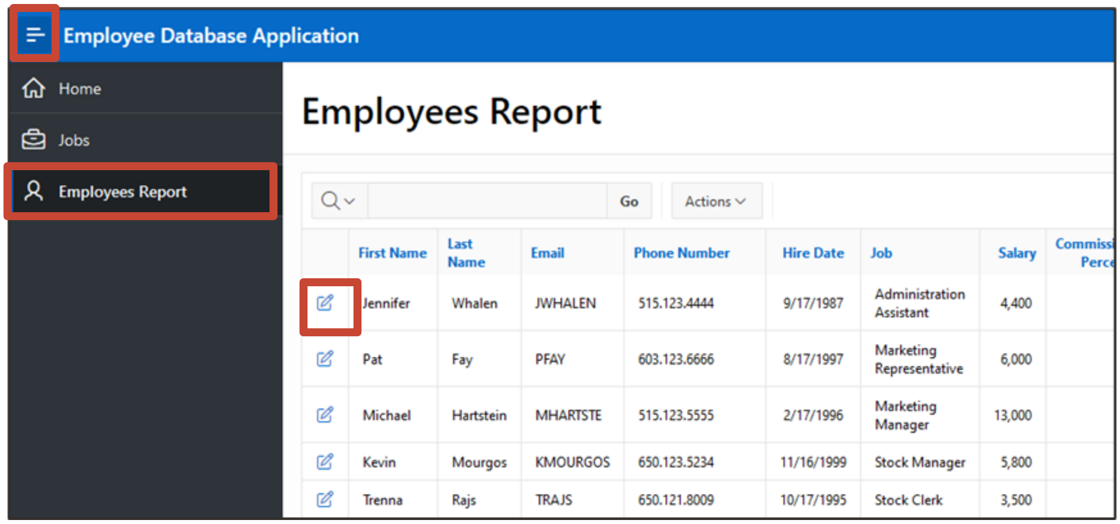
2. The Employees Report will open in the **Page Designer** view. Click the **Application *Number*** to return to the application's home page. (**Note**: Your application number may look different.)



3. You will see that two new pages have been added to your app, Employees report and Update Employees. Click the **Run Application** icon.



4. Login with your APEX username and password if prompted, click **Employees Report** from the **Navigation Menu**, and you will see all employees displayed. Click the edit icon in a row of data.



	First Name	Last Name	Email	Phone Number	Hire Date	Job	Salary	Commissi Perce
✎	Jennifer	Whalen	JWHALEN	515.123.4444	9/17/1987	Administration Assistant	4,400	
✎	Pat	Fay	PFAY	603.123.6666	8/17/1997	Marketing Representative	6,000	
✎	Michael	Hartstein	MHARTSTE	515.123.5555	2/17/1996	Marketing Manager	13,000	
✎	Kevin	Mourgos	KMOURGOS	650.123.5234	11/16/1999	Stock Manager	5,800	
✎	Trenna	Rajs	TRAJS	650.121.8009	10/17/1995	Stock Clerk	3,500	

5. The Update Employees Form is displayed, which allows employee details to be edited. Do not attempt to modify or insert records at this time. Close the Update Employee form by either clicking one of the pages in the Navigation Menu on the left of the window or scroll down the Update Employee form and click the Cancel button.

Part 3: Include Lists of Values on the Update Employees page

The employees table has three foreign key columns. In order to successfully insert or update a record, valid values must be used for these columns that already exist in the parent tables. Using values that are not present in the parent tables will result in an integrity constraint error when attempting to insert or update a record.

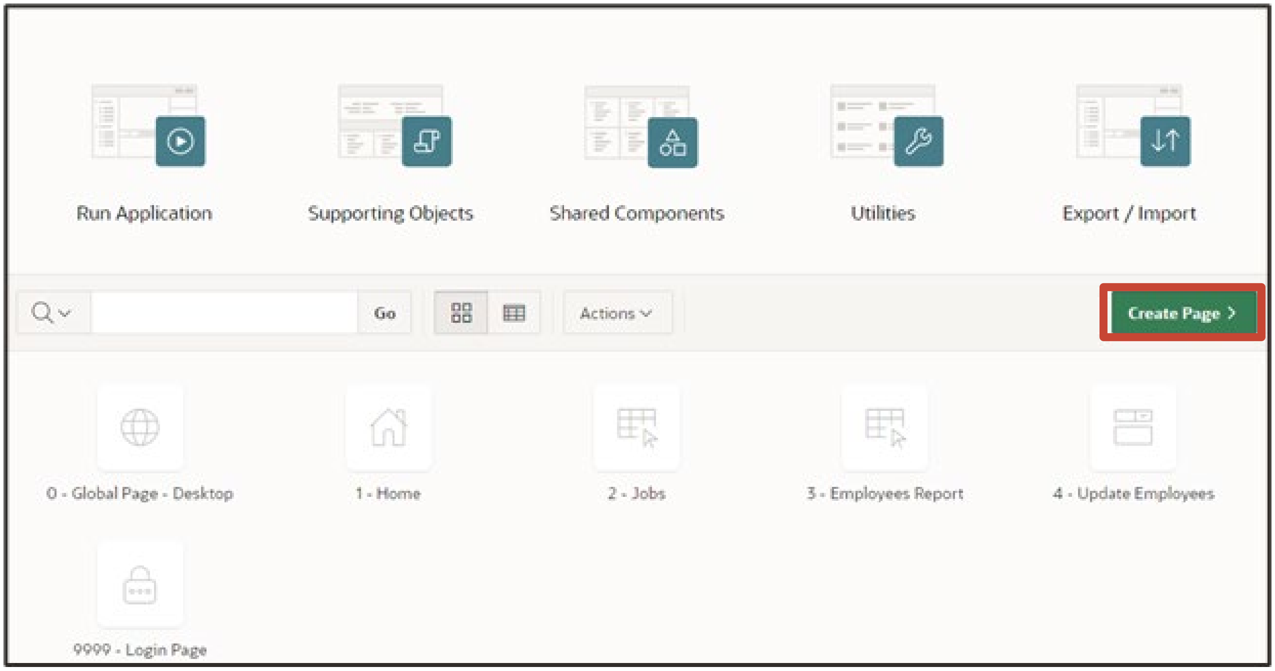
By default, the **Job Id**, **Manager Id**, and **Department Id** fields are of type **Select List**, which use an automatically created **List of Values** to allow the user to select existing values from the parent table.

The **Manager Id**, **Select List**, picks up all employees regardless of whether they are manager or not, and only displays the first name.

To address this, we can create a **List of Values** (LOVs) manually, that forces the user to select only valid values, employees that are managers. We can also display both first and last name.

Step 1: Open the Update Employees Page for Editing

1. From the application home page, Click page **4 – Update Employees**.

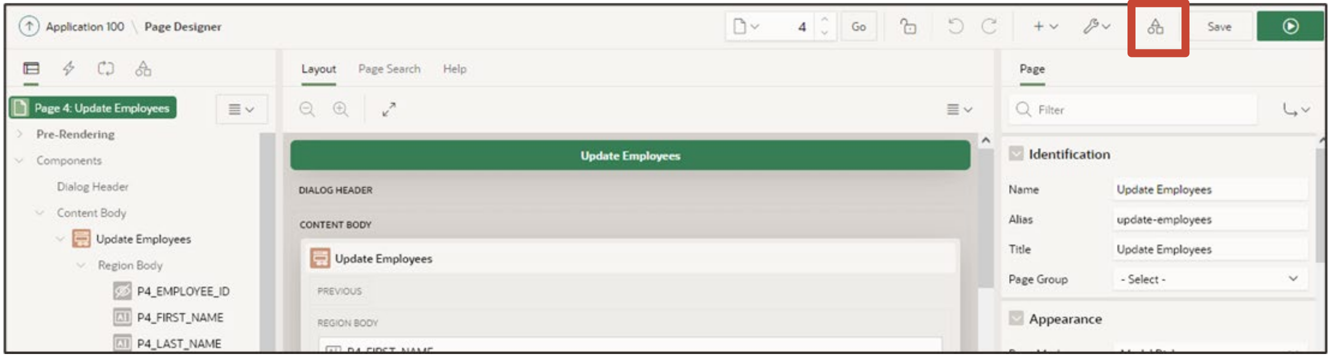


2. The page will open in **Page Designer** view.

Step 2: Create a Shared List of Values

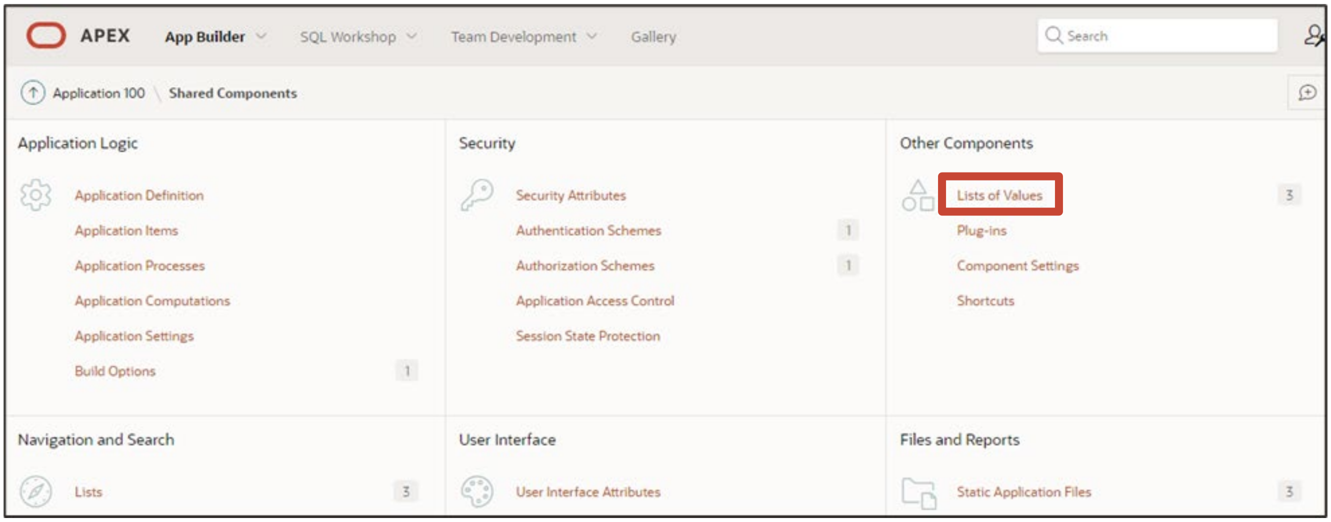
We will now create a new **List of Values** (LOV) to use for the `manager_id` field. Later in this Lab, we will create a page to view and modify the `departments` table, which also has a `manager_id` column. To save duplication of work, we can create the LOV as a **Shared Component**, and access it from both pages.

1. From the Page Designer, click the **Shared Components** icon.

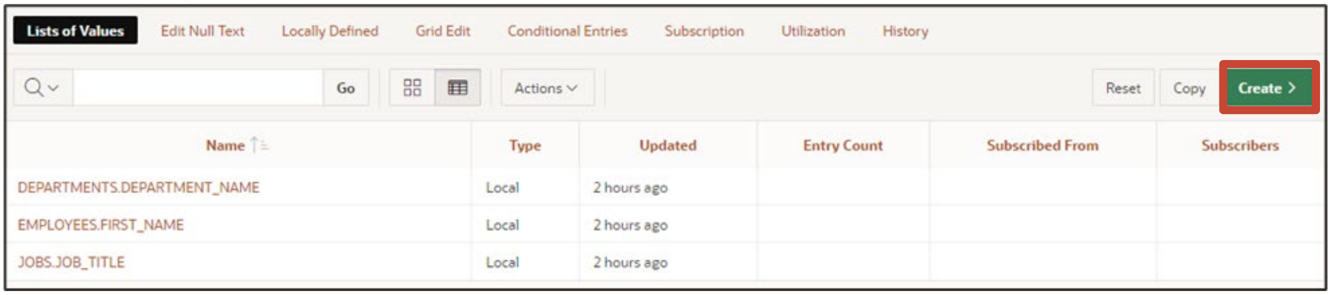


Name: Update Employees
Alias: update-employees
Title: Update Employees
Page Group: - Select -

2. In the **Other Components** section, click **Lists of Values**.

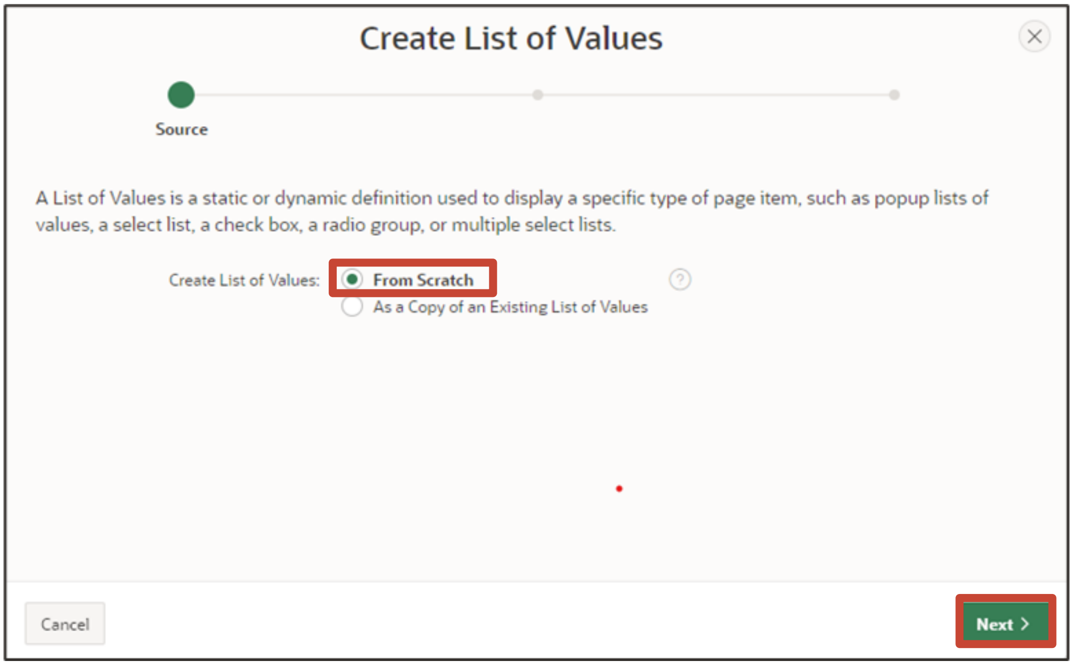


3. To create an LOV manually, click the **Create** button. (**Note**: The 3 existing LOVs that were automatically created.)



Name	Type	Updated	Entry Count	Subscribed From	Subscribers
DEPARTMENTS.DEPARTMENT_NAME	Local	2 hours ago			
EMPLOYEES.FIRST_NAME	Local	2 hours ago			
JOBS.JOB_TITLE	Local	2 hours ago			

4. For **Source**, select **From Scratch**, then click **Next**.



A List of Values is a static or dynamic definition used to display a specific type of page item, such as popup lists of values, a select list, a check box, a radio group, or multiple select lists.

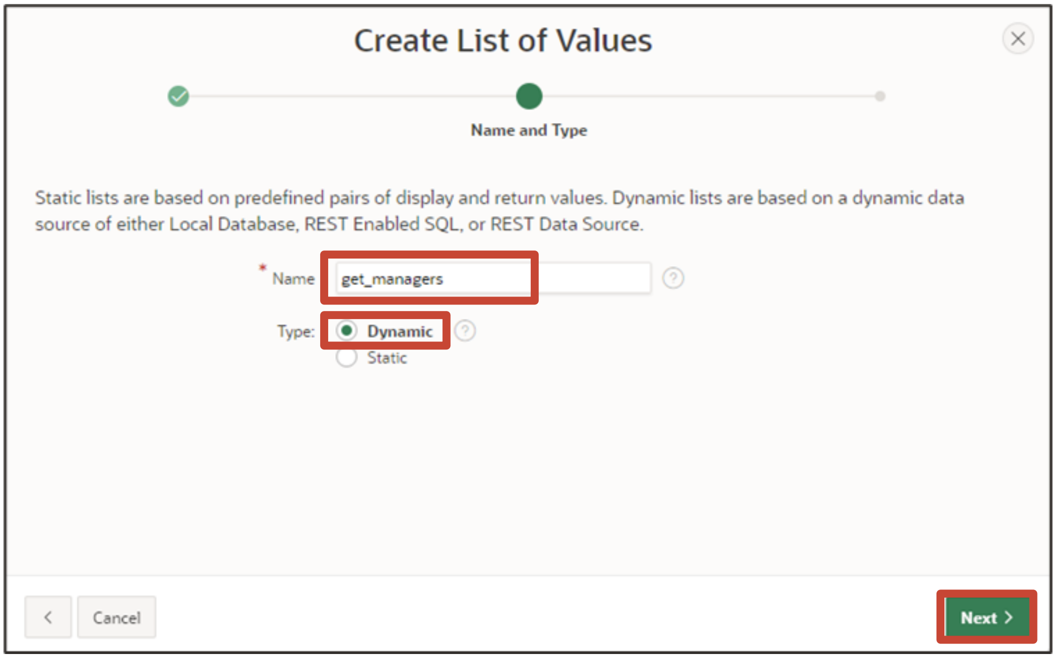
Create List of Values: ● From Scratch
○ As a Copy of an Existing List of Values

5. Enter the following:

   Name: `get_managers`

   Type: Dynamic

   Click Next.



Create List of Values

Name and Type

Static lists are based on predefined pairs of display and return values. Dynamic lists are based on a dynamic data source of either Local Database, REST Enabled SQL, or REST Data Source.

* Name: get_managers

Type: ● Dynamic ○ Static

6. The SQL statement for this LOV has two functions to complete. Firstly, we want to display both the manager first name and last name. As these are two different columns in the table, we need to use the concatenation operator (two “pipe” symbols `||`) to display the manager full name as if it were a single column.

   We also need to add a `WHERE` clause as, only employees with a job id of `AD_PRES`, `AD_VP` or employees with the text `_MAN` or `_MGR` in their `job_id` can be managers.

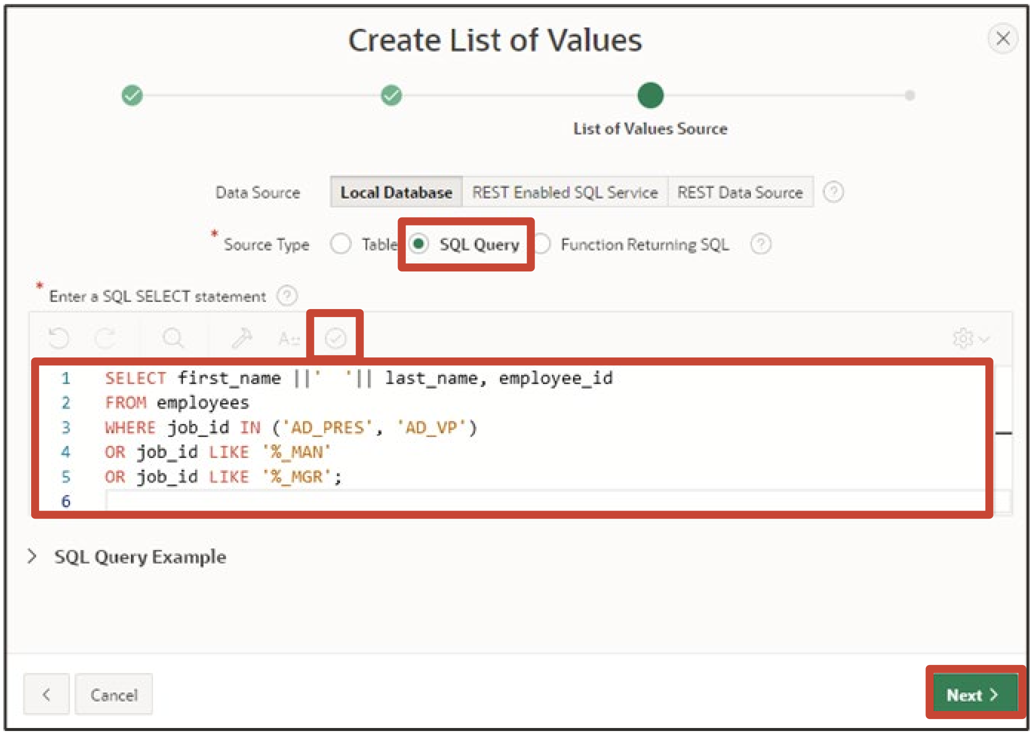
   For **Source Type**: Select **SQL Query**

   Enter the following code in the **Code Editor**:

```
SELECT first_name ||'  '|| last_name, employee_id
FROM employees
WHERE job_id IN ('AD_PRES', 'AD_VP')
OR job_id LIKE '%_MAN'
OR job_id LIKE '%_MGR';
```

ORACLE Academy

7. Once the statement is entered, click the **Validation** icon (a tick inside a circle). Once you receive a validation successful message, click **Next**.



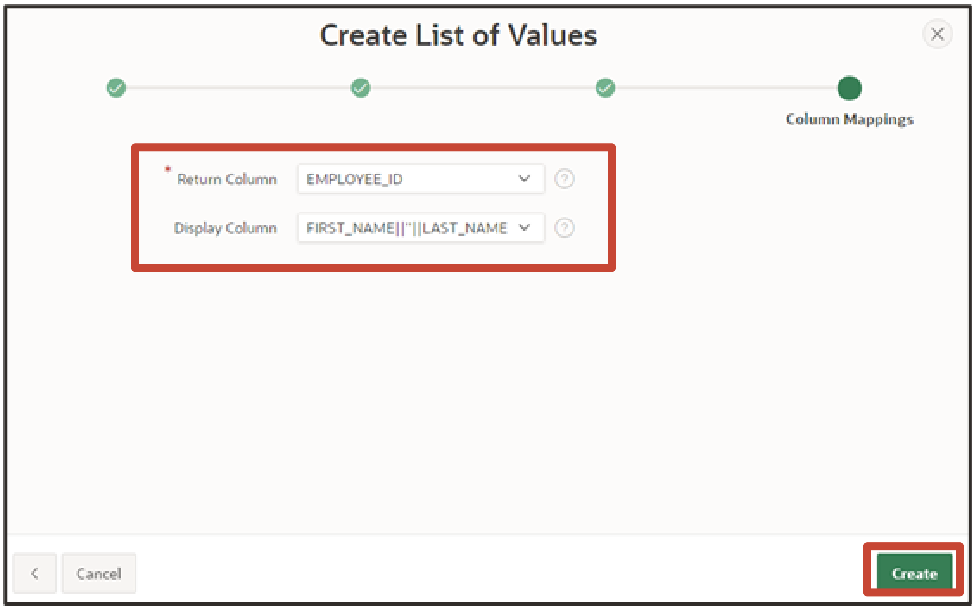
Data Source: Local Database | REST Enabled SQL Service | REST Data Source

* Source Type ○ Table ● SQL Query ○ Function Returning SQL

* Enter a SQL SELECT statement

```
SELECT first_name ||'  '|| last_name, employee_id
FROM employees
WHERE job_id IN ('AD_PRES', 'AD_VP')
OR job_id LIKE '%_MAN'
OR job_id LIKE '%_MGR';
```

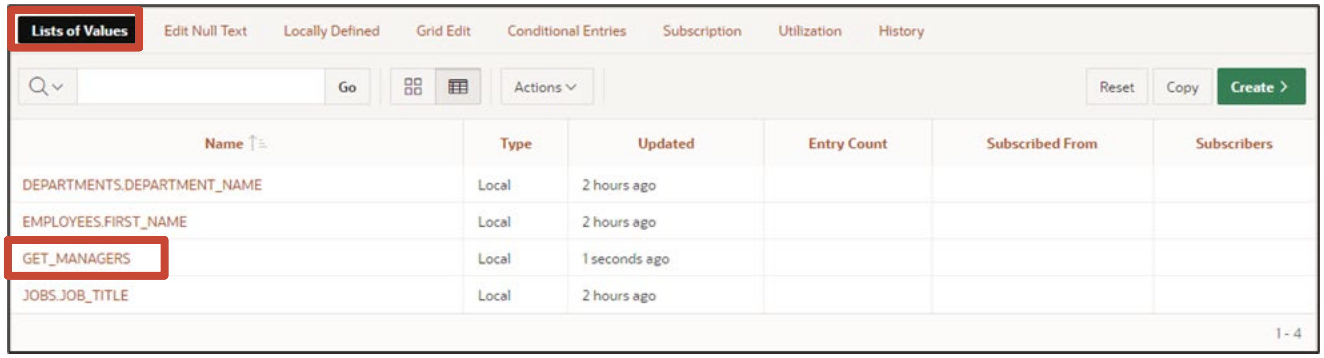
8. Accept the values for Column Mappings and click Create.



* Return Column: EMPLOYEE_ID

Display Column: FIRST_NAME||''||LAST_NAME

9. The **Lists of Values** will be displayed on the Shared Component's LOV page. Click the **Application *Number*** to return to the application home page.



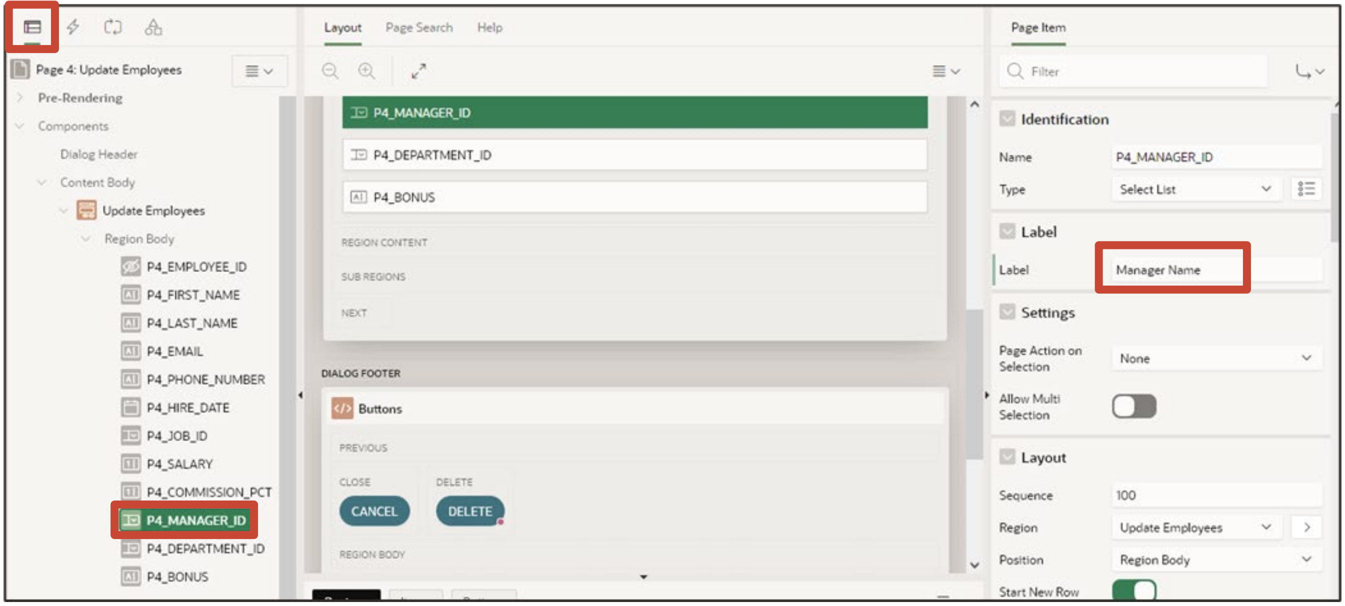
Name	Type	Updated	Entry Count	Subscribed From	Subscribers
DEPARTMENTS.DEPARTMENT_NAME	Local	2 hours ago			
EMPLOYEES.FIRST_NAME	Local	2 hours ago			
GET_MANAGERS	Local	1 seconds ago			
JOBS.JOB_TITLE	Local	2 hours ago			

Step 3: Use the Shared List of Values in the Update Employees Page

From the application home page, click page **4 – Update Employees** to open the page in Page Designer.

1. From the Rendering tab, select `P4_MANAGER_ID` by left mouse clicking it. (**Note**: If the items are not listed, you may need to expand **Components**, **Content Body** and **Region Body**).

   In the **Page Item** area, change the **Label** to **Manager Name**.



Identification

Name: P4_MANAGER_ID

Type: Select List

Label

Label: Manager Name

Settings

Page Action on Selection: None

Allow Multi Selection

Layout

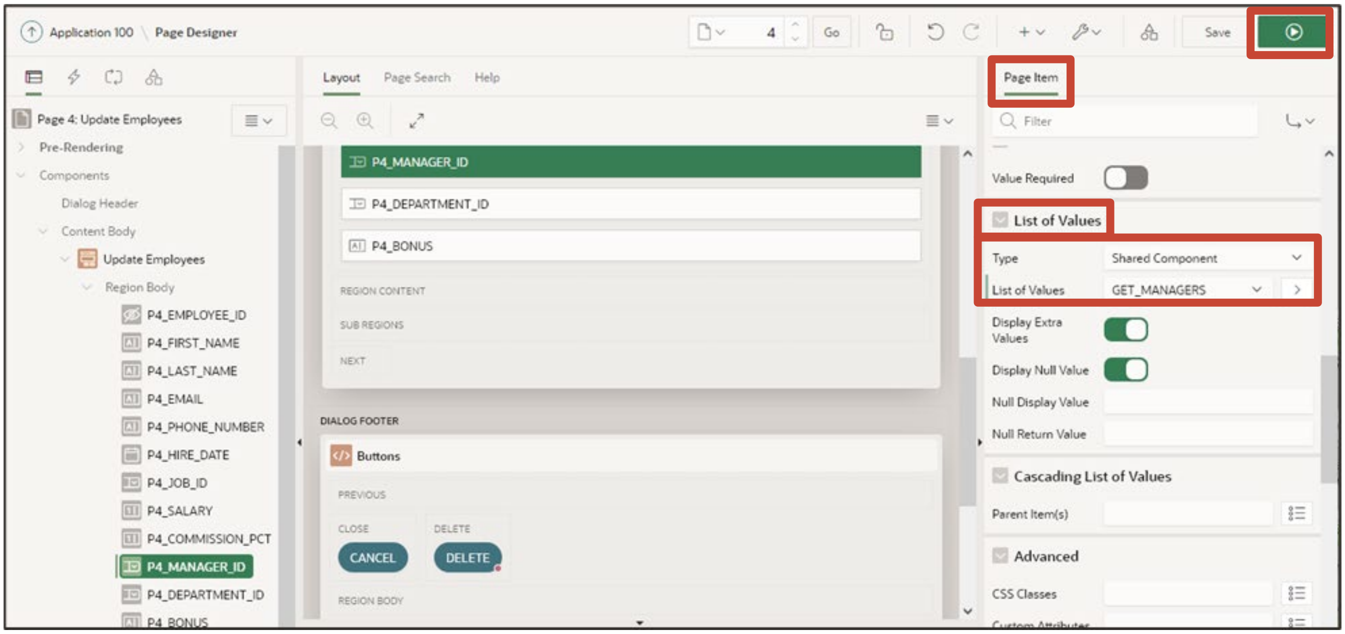
Sequence: 100

Region: Update Employees

Position: Region Body

Start New Row

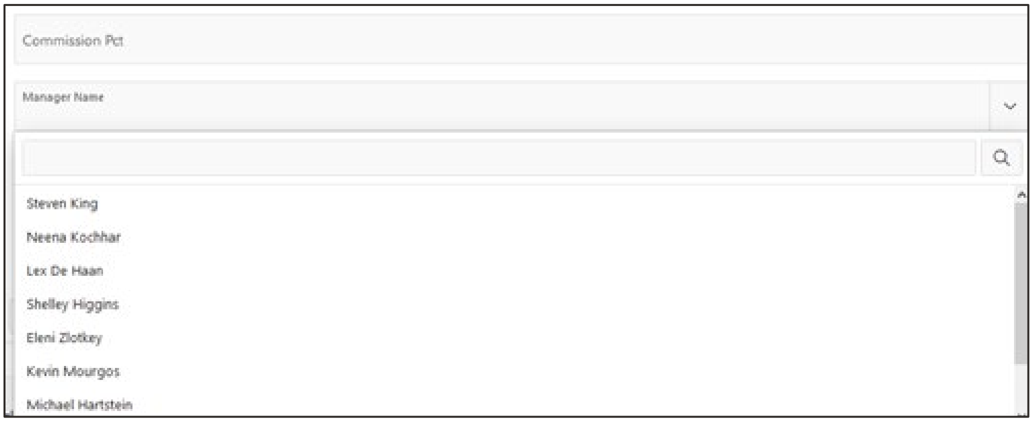
2. Continuing in the **Page Item** area, scroll down to see the **List of Values** settings. For **Type**, select **Shared Component** and for **List of Values**, select **GET_MANAGERS**. Click **Save**.



Type: Shared Component

List of Values: GET_MANAGERS

3. Run the application to test. If you have coded your SQL statement correctly, the LOV for **Manager Name** should show only the seven employees that can be managers. (**Note**: Do not attempt to insert a new record at this time.)



Steven King
Neena Kochhar
Lex De Haan
Shelley Higgins
Eleni Zlotkey
Kevin Mourgos
Michael Hartstein

Part 4: Use a Sequence to Manage Primary Key Values on the Update Employees Page

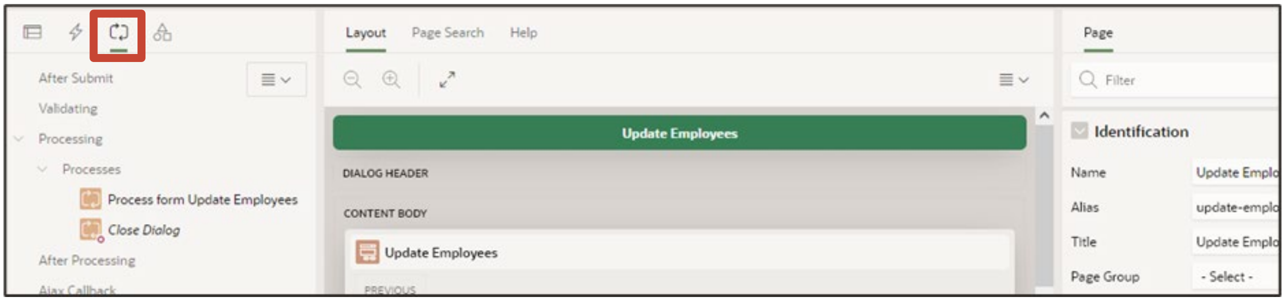
If you tried to insert a new employee on the **Update Employee** page, you would receive an error message. The `employee_id` item is hidden by default, so the user cannot enter a value for the primary key of the table. One solution would be to edit the page and change the `employee_id` item's type from **Hidden** to **Number Field**, and users could then enter a value for the `employee_id`.

It is, however, unrealistic to expect end users to know what value to use for the primary key, especially if the table is large and has many rows.

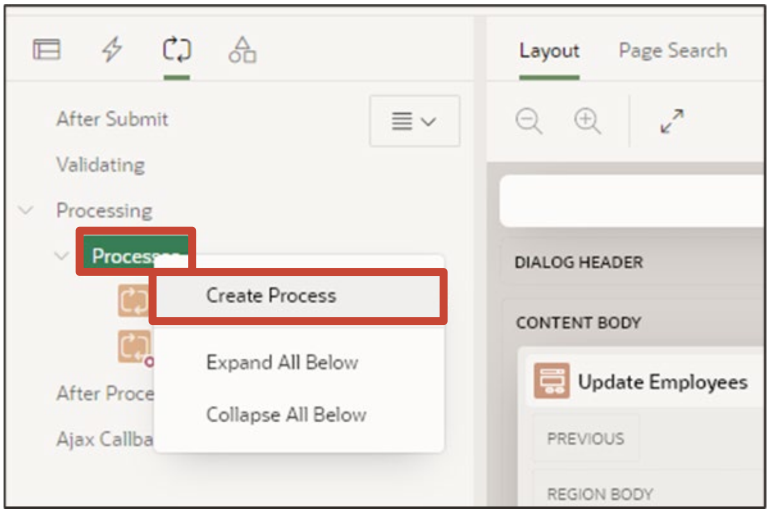
We can create a process that will use a sequence, `employees_seq` (which was created in your schema when you ran the script to create and populate the tables) to automatically add values for the `employee_id` column in the table.

Step 1: Create a New Process

1. Open page **4 - Update Employees** in Page Designer view as before and click the **Processing** tab.



2. Right click on **Processes** and select **Create Process**.



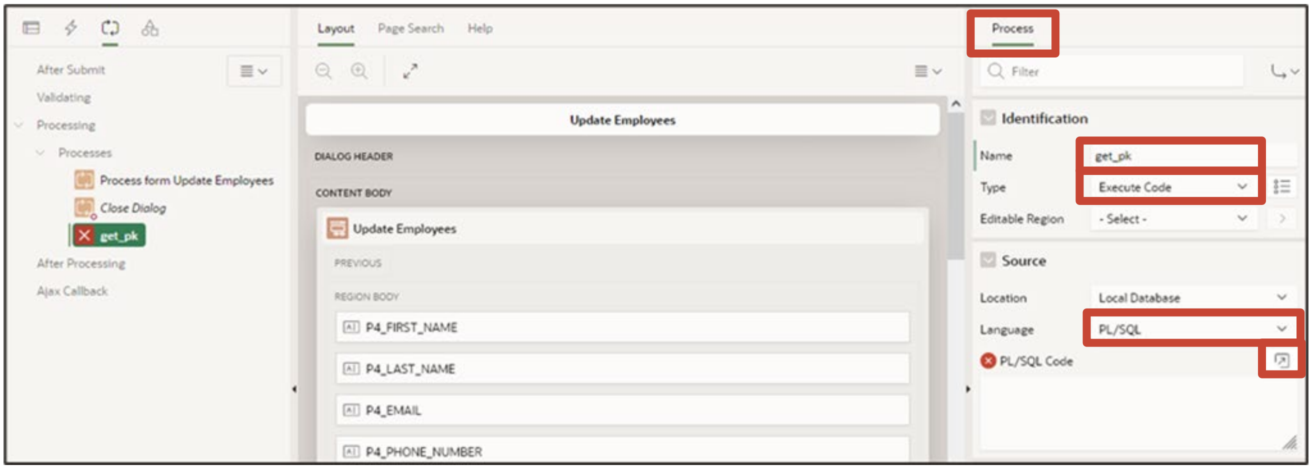
Step 2: Enter Code for the Process

1. Enter the following for **Process**. In the **Identification** area, enter:

   Name: `get_pk`

   Type: Execute Code

   In the **Source** area, set **Language** to PL/SQL, then click the **Code editor** icon.



Name: get_pk

Type: Execute Code

Editable Region: - Select -

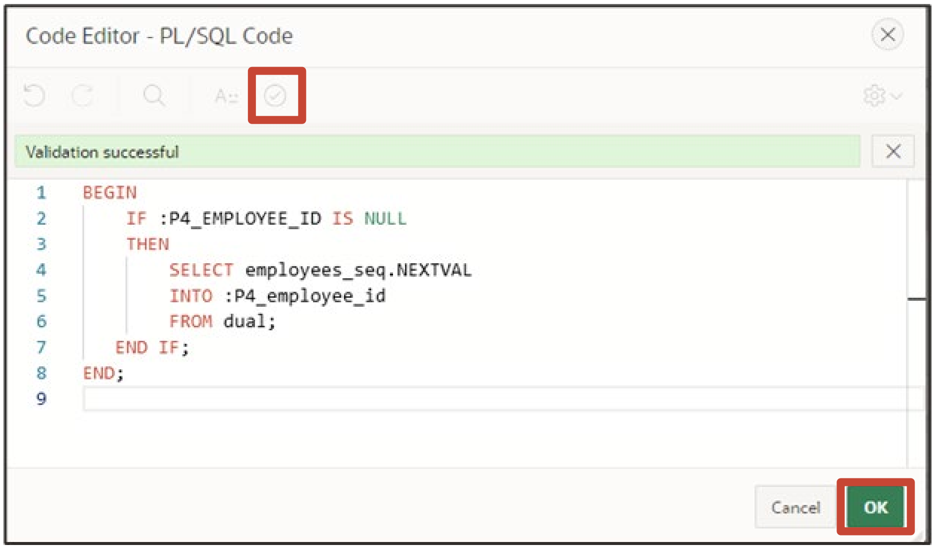
Location: Local Database

Language: PL/SQL

2. Enter the following code in the code editor:

```
BEGIN
    IF :P4_EMPLOYEE_ID IS NULL
    THEN
        SELECT employees_seq.NEXTVAL
        INTO :P4_employee_id
        FROM dual;
   END IF;
END;
```

3. Once the statement is entered, click the **Validation** icon (a tick inside a circle), and if you receive a validation successful message, click **OK**.

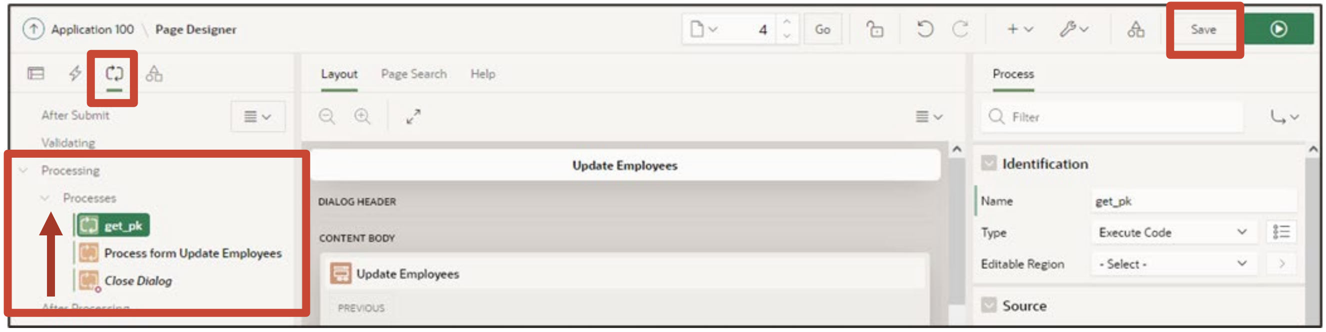


Validation successful

```
BEGIN
    IF :P4_EMPLOYEE_ID IS NULL
    THEN
        SELECT employees_seq.NEXTVAL
        INTO :P4_employee_id
        FROM dual;
   END IF;
END;
```

4. The new `get_pk` process must run before the other process to ensure that `employee_id` has a value before the insert is processed.

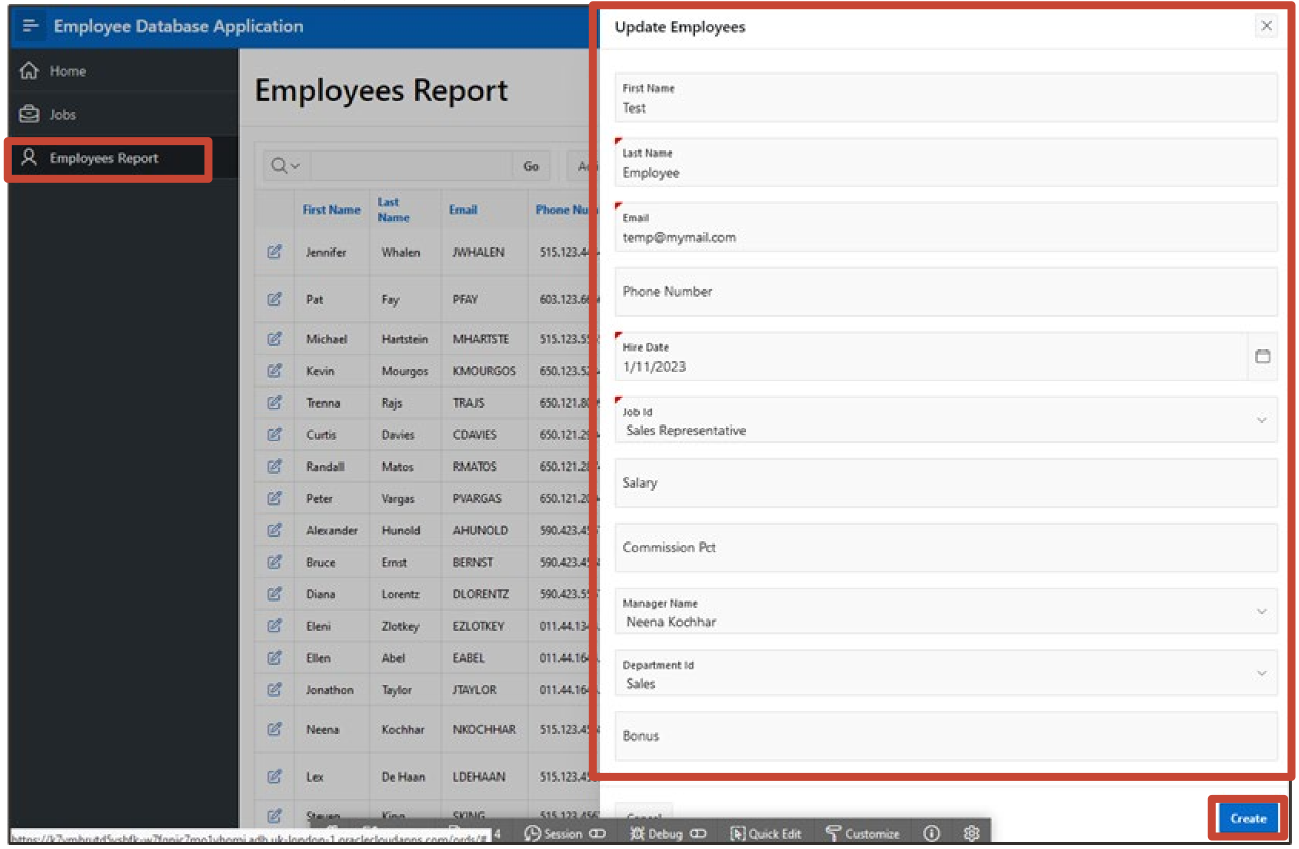
5. In the processes tab, left mouse click and drag the new `get_pk` process so that it is above the **Process form Update Employees** process. Click **Save**.



Name: get_pk
Type: Execute Code
Editable Region: - Select -

Step 3: Run and Test

1. Run the application to test. Open the **Employees Report** and click the **Create** button. Enter values for a new employee and click **Create**.



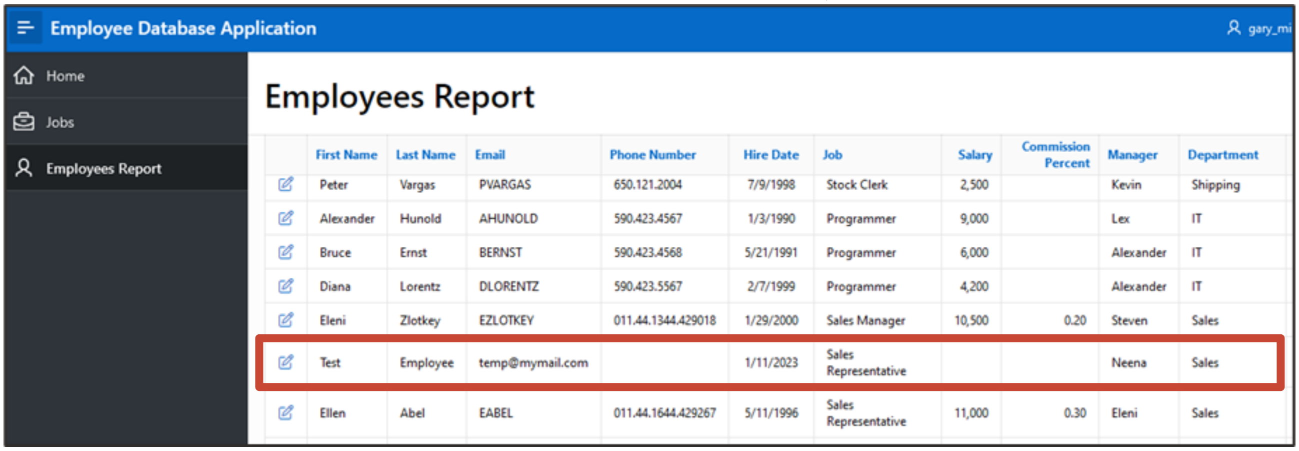
Employees Report

	First Name	Last Name	Email	Phone Nu
	Jennifer	Whalen	JWHALEN	515.123.4
	Pat	Fay	PFAY	603.123.6
	Michael	Hartstein	MHARTSTE	515.123.5
	Kevin	Mourgos	KMOURGOS	650.123.5
	Trenna	Rajs	TRAJS	650.121.8
	Curtis	Davies	CDAVIES	650.121.2
	Randall	Matos	RMATOS	650.121.2
	Peter	Vargas	PVARGAS	650.121.2
	Alexander	Hunold	AHUNOLD	590.423.4
	Bruce	Ernst	BERNST	590.423.4
	Diana	Lorentz	DLORENTZ	590.423.5
	Eleni	Zlotkey	EZLOTKEY	011.44.13
	Ellen	Abel	EABEL	011.44.16
	Jonathon	Taylor	JTAYLOR	011.44.16
	Neena	Kochhar	NKOCHHAR	515.123.4
	Lex	De Haan	LDEHAAN	515.123.4
	Steven	King	SKING	515.123.456

Update Employees

First Name	Test
Last Name	Employee
Email	temp@mymail.com
Phone Number	
Hire Date	1/11/2023
Job Id	Sales Representative
Salary	
Commission Pct	
Manager Name	Neena Kochhar
Department Id	Sales
Bonus	

2. You should see the new employee included in the **Employees Report** page.



Employees Report

	First Name	Last Name	Email	Phone Number	Hire Date	Job	Salary	Commission Percent	Manager	Department
	Peter	Vargas	PVARGAS	650.121.2004	7/9/1998	Stock Clerk	2,500		Kevin	Shipping
	Alexander	Hunold	AHUNOLD	590.423.4567	1/3/1990	Programmer	9,000		Lex	IT
	Bruce	Ernst	BERNST	590.423.4568	5/21/1991	Programmer	6,000		Alexander	IT
	Diana	Lorentz	DLORENTZ	590.423.5567	2/7/1999	Programmer	4,200		Alexander	IT
	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	1/29/2000	Sales Manager	10,500	0.20	Steven	Sales
	Test	Employee	temp@mymail.com		1/11/2023	Sales Representative			Neena	Sales
	Ellen	Abel	EABEL	011.44.1644.429267	5/11/1996	Sales Representative	11,000	0.30	Eleni	Sales

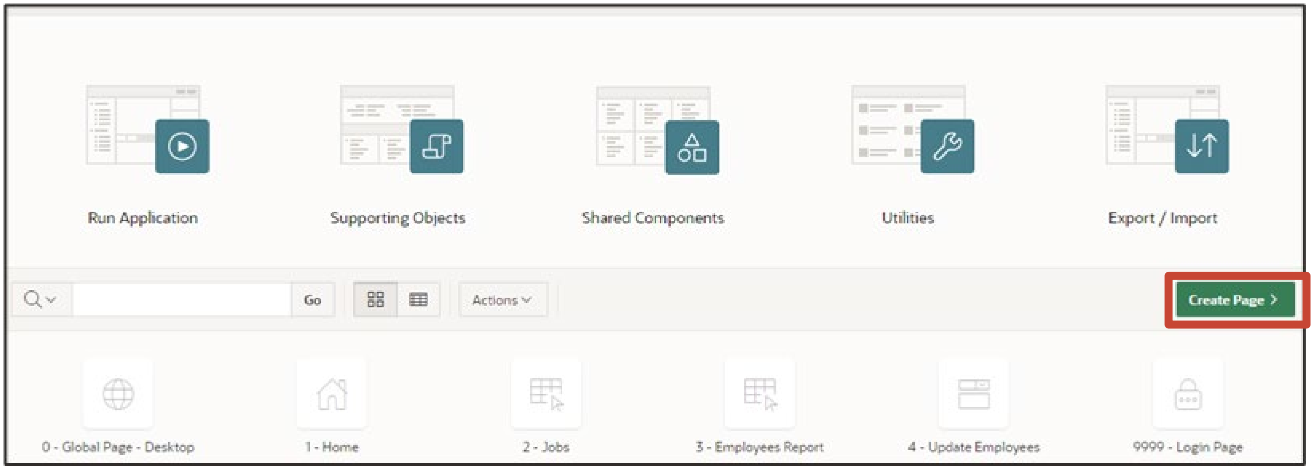
Part 5: Add a New Page to View Departments and their Employees

In this section, we will add a **Master Detail** form, which will display departments, and the employees that are assigned to each department on a separate page. The page will also allow the user to update, add and delete departments.

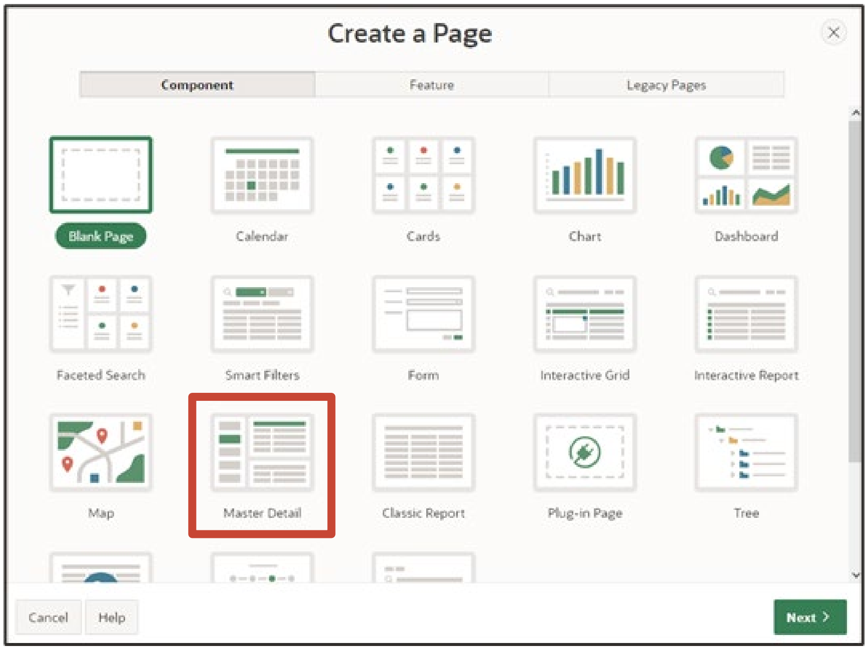
Master Detail refers to the relationship between the two tables. The `employees` table has a foreign key relationship with the `departments` table, so the `departments` table is known as the **master** table, and `employees` as the **detail** table.

Step 1: Run the Page Wizard to Create the New Forms

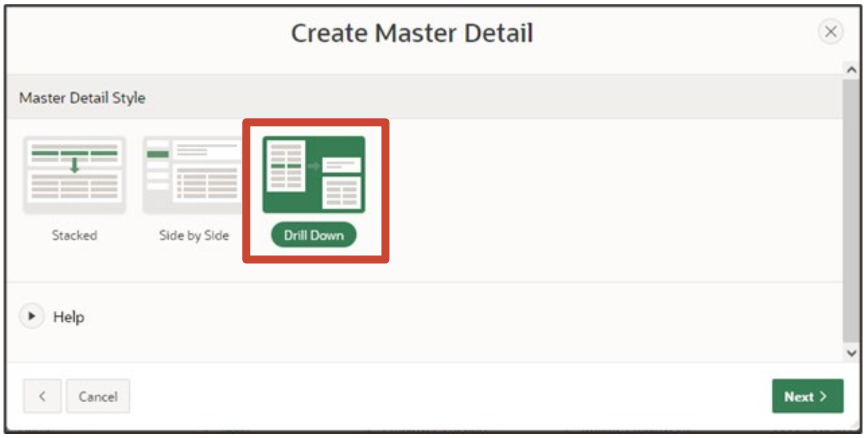
1. From the application home page, click the **Create Page >** button.



2. Select **Master Detail**.



3. Select **Drill Down**, click **Next**.



Create Master Detail

Master Detail Style

4. For **Master Page Definition**, enter:

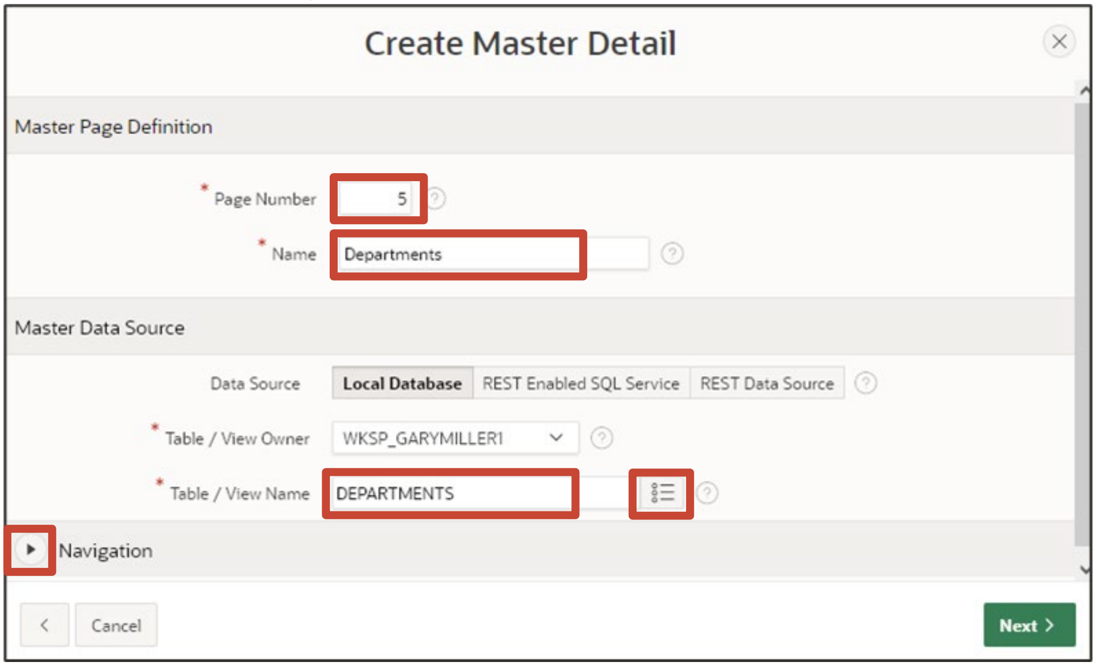
   Page Number: 5

   Name: `Departments`

   For **Master Data Source**,

   Table / View Name: Click the list icon and select `DEPARTMENTS`

   Click to expand **Navigation**.



Create Master Detail

Master Page Definition

* Page Number: 5

* Name: Departments

Master Data Source

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

* Table / View Owner: WKSP_GARYMILLER1

* Table / View Name: DEPARTMENTS

Step 2: Apply Navigation Menu Settings

1. Scroll down to the **Navigation** properties and check the following properties are selected:

   Use Breadcrumb: Move the slider to make it green (on position)

   Breadcrumb Parent Entry: empty

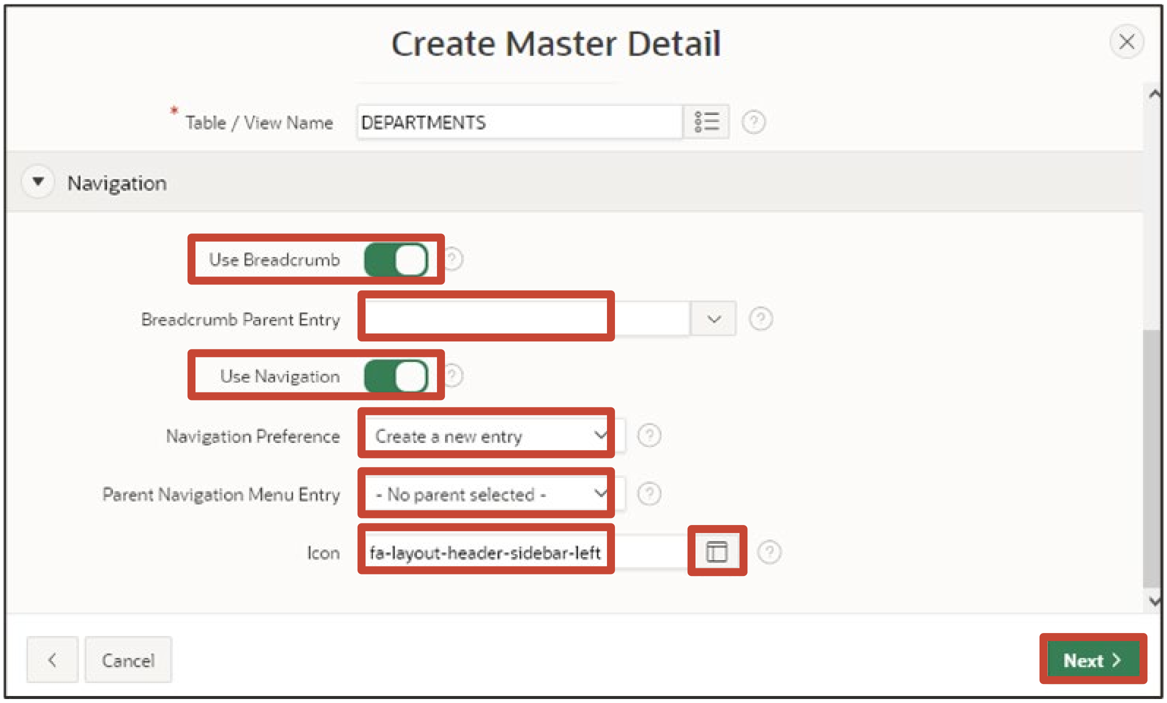
   Use Navigation: Move the slider to make it green (on position)

   Navigation Preference: Create a new entry

   Parent Navigation Menu Entry: - No Parent Selected -

   You can select a page **icon** by clicking the **Select Icon** button.

   Click the **Next** button.



Create Master Detail

* Table / View Name: DEPARTMENTS

Navigation

Use Breadcrumb: on

Breadcrumb Parent Entry:

Use Navigation: on

Navigation Preference: Create a new entry

Parent Navigation Menu Entry: - No parent selected -

Icon: fa-layout-header-sidebar-left

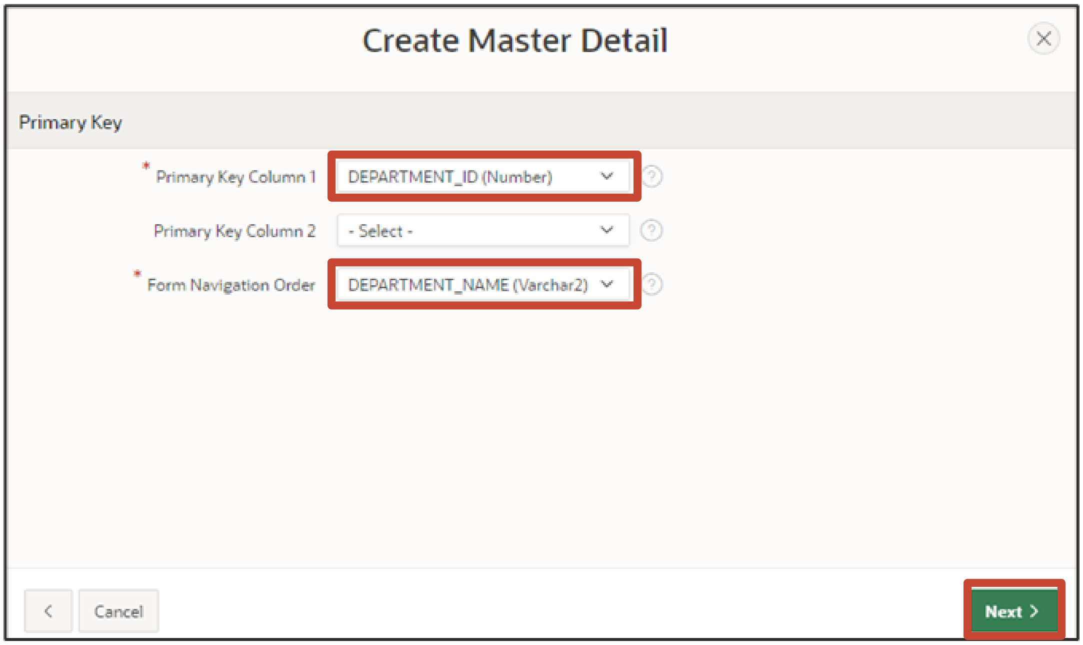
Step 3: Select Primary Key

1. Check the following properties for **Primary Key**:

   Primary Key Column 1: `DEPARTMENT_ID` (Number)

   Form Navigation Order: `DEPARTMENT_NAME` (Varchar2)

   Click **Next**.



Create Master Detail

Primary Key

* Primary Key Column 1: DEPARTMENT_ID (Number)

Primary Key Column 2: - Select -

* Form Navigation Order: DEPARTMENT_NAME (Varchar2)

Step 4: Select Detail Table and Primary Key

1. Check the following properties for **Detail Page Definition**:

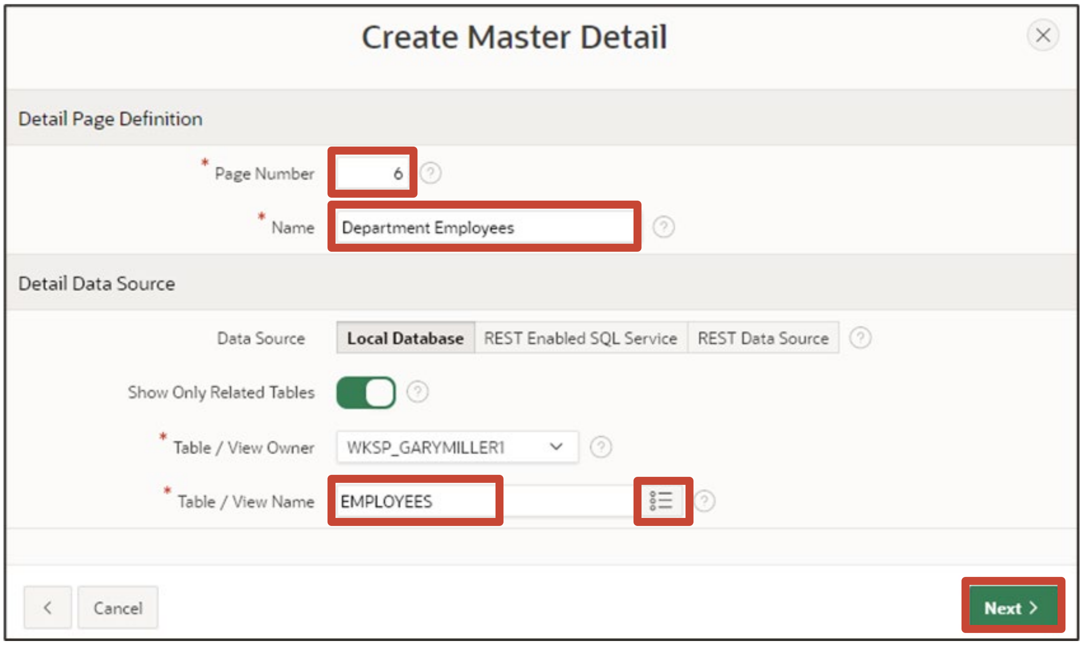
   Page Number: 6

   Name: Department Employees

   Check the following properties for **Detail Data Source**:

   Table / View: `EMPLOYEES`

   Click **Next**.



Create Master Detail

Detail Page Definition

* Page Number: 6

* Name: Department Employees

Detail Data Source

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

Show Only Related Tables: On

* Table / View Owner: WKSP_GARYMILLER1

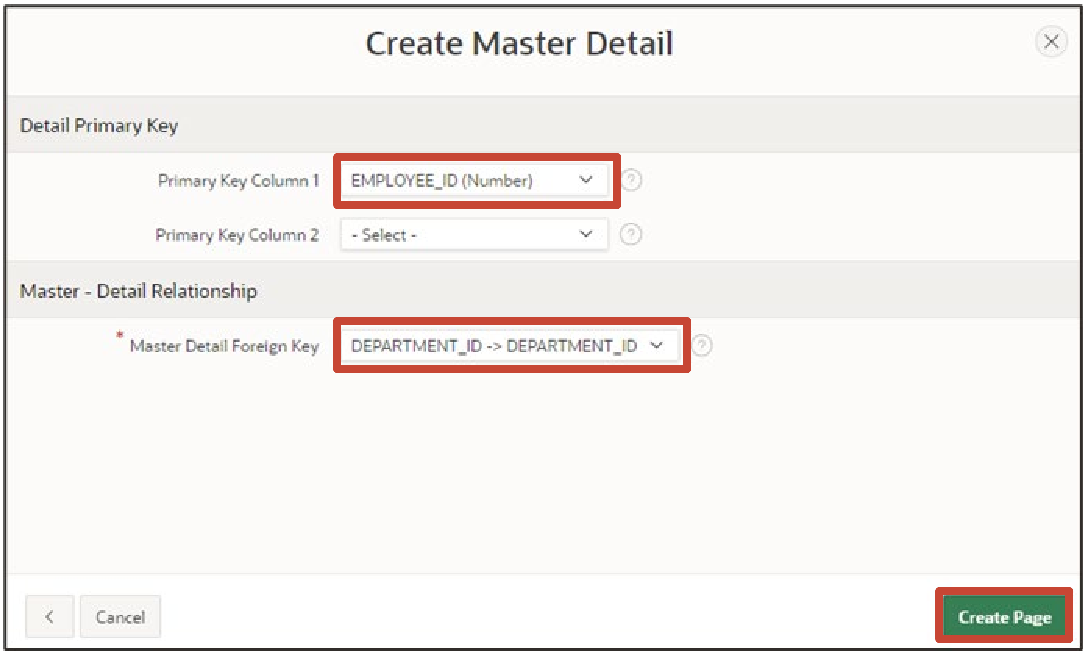
* Table / View Name: EMPLOYEES

2. Check the following **Detail** properties:

   Primary Key Column 1: `EMPLOYEE_ID` (Number)

   Master Detail Relationship: `DEPARTMENT_ID` -> `DEPARTMENT_ID`

   Click **Create Page**.



Create Master Detail

Detail Primary Key

Primary Key Column 1: EMPLOYEE_ID (Number)

Primary Key Column 2: - Select -

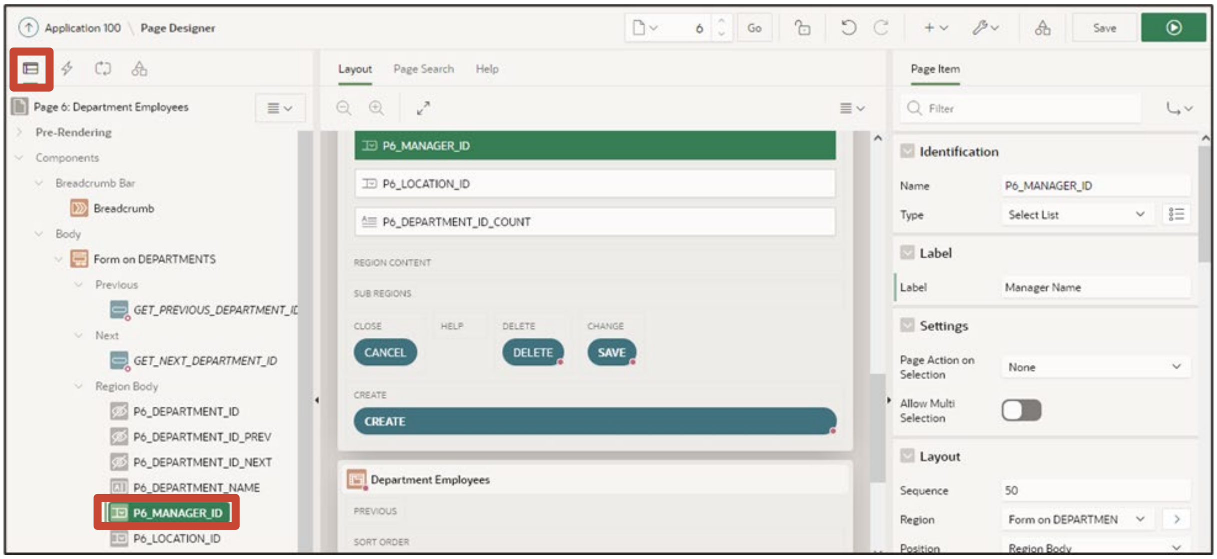
Master - Detail Relationship

* Master Detail Foreign Key: DEPARTMENT_ID -> DEPARTMENT_ID

Step 5: Use LOV for `manager_id` Foreign Key Column

Ensure page **6 – Department Employees** is open in **Page Designer**.

1. From the Rendering tab, select **P6_MANAGER_ID** by left mouse clicking it. (If the items are not listed, you may need to expand **Components**, **Body**, **Form on Departments** and **Region Body**).



Identification

Name: P6_MANAGER_ID

Type: Select List

Label

Label: Manager Name

Settings

Page Action on Selection: None

Allow Multi Selection

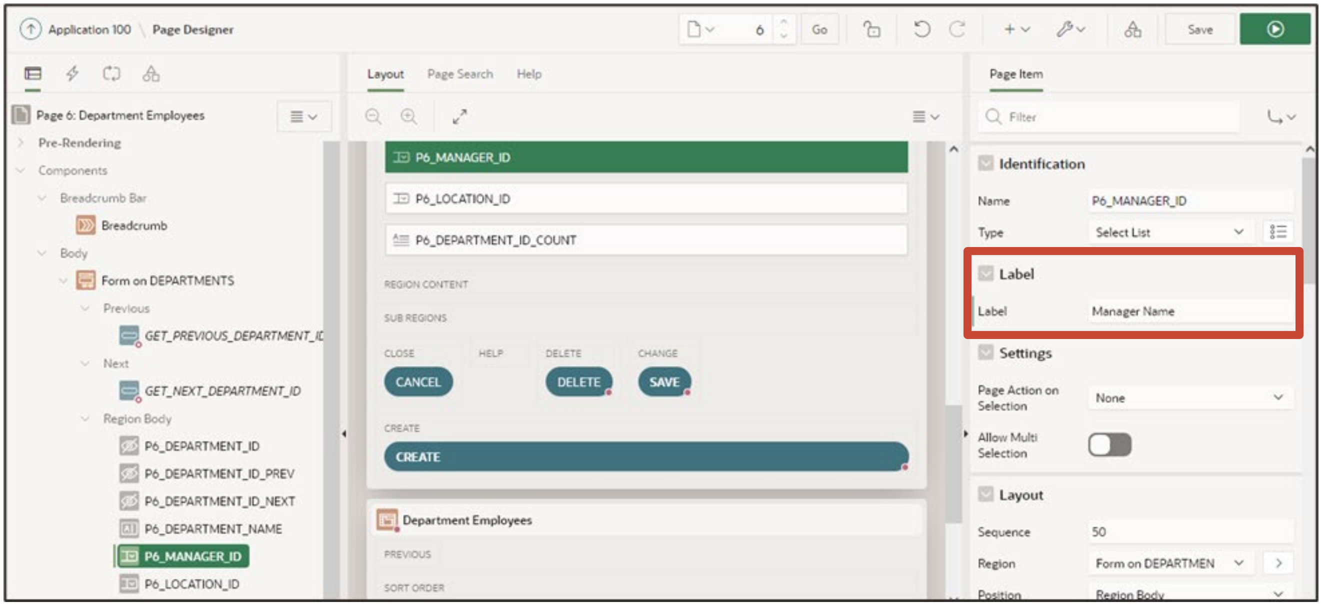
Layout

Sequence: 50

Region: Form on DEPARTMEN

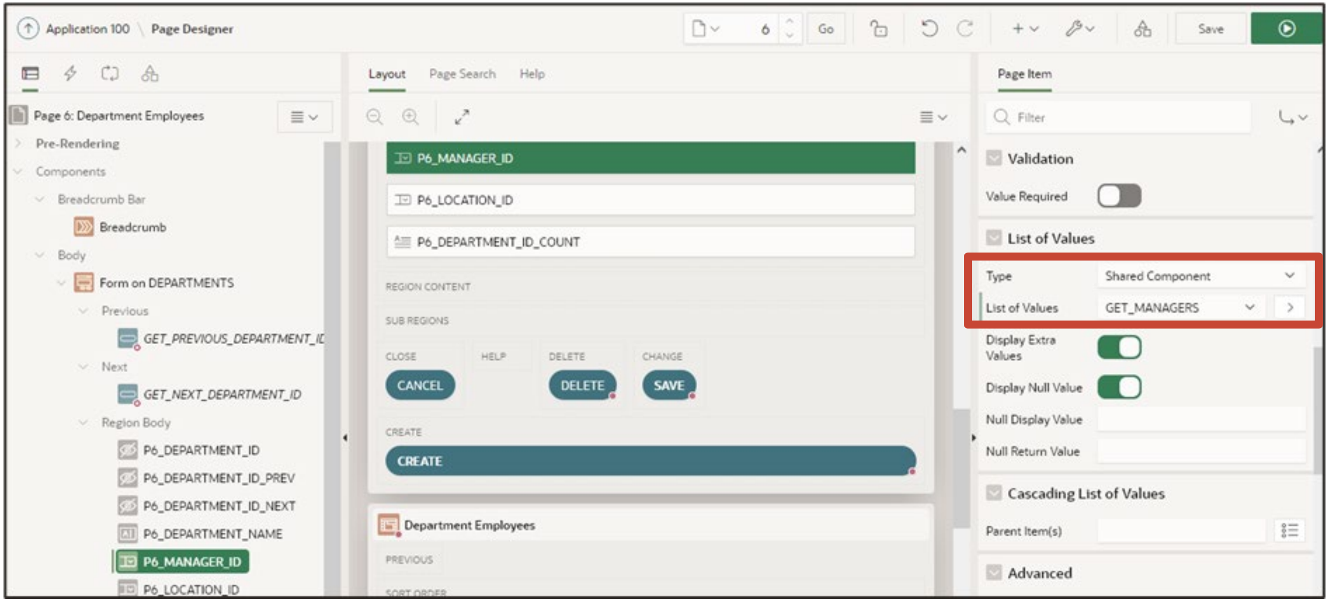
Position: Region Body

2. In the **Page Item** area, change the **Label** to **Manager Name**.



Name: P6_MANAGER_ID
Type: Select List
Label
Label: Manager Name
Page Action on Selection: None
Allow Multi Selection
Sequence: 50
Region: Form on DEPARTMEN
Position: Region Body

3. In the **Page Item** area, scroll down to see the **List of Values** settings. For **Type**, select **Shared Component**, for **List of Values**, select **`GET_MANAGERS`** (which we created earlier). Click **Save**.



Value Required
List of Values
Type: Shared Component
List of Values: GET_MANAGERS
Display Extra Values
Display Null Value
Null Display Value
Null Return Value

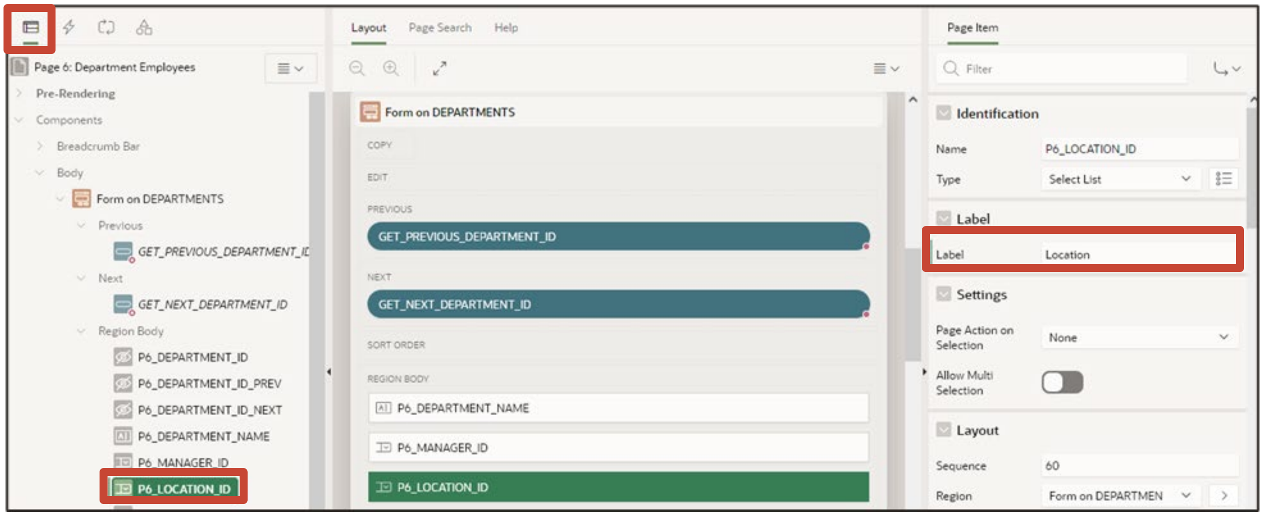
Step 6: Edit LOV for `location_id` Foreign Key Column

By default, the `location_id` foreign key uses the automatically generated shared LOV that displays street address. In this step, we will change this to display both `city` and `street`, using a SQL statement instead of the default LOV.

Ensure page **6 – Department Employees** is open in **Page Designer**.

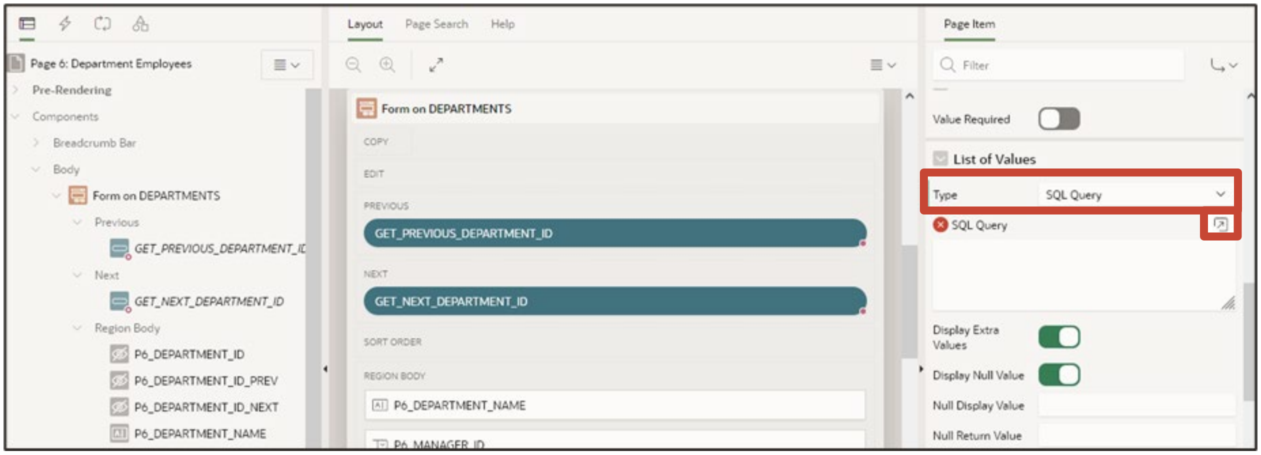
1. From the rendering tab, select **P6_LOCATION_ID** by left mouse clicking it. (If the items are not listed, you may need to expand **Components**, **Body**, **Form on Departments**, **Region Body**).

   In the **Page Item** area, change the **Label** to **Location**.



Name: P6_LOCATION_ID
Type: Select List
Label: Location
Page Action on Selection: None
Allow Multi Selection
Sequence: 60
Region: Form on DEPARTMEN

2. In the **Page Item** area, scroll down to see the **List of Values** settings. For **Type**, select **SQL Query** and click the **icon** to open the code editor:

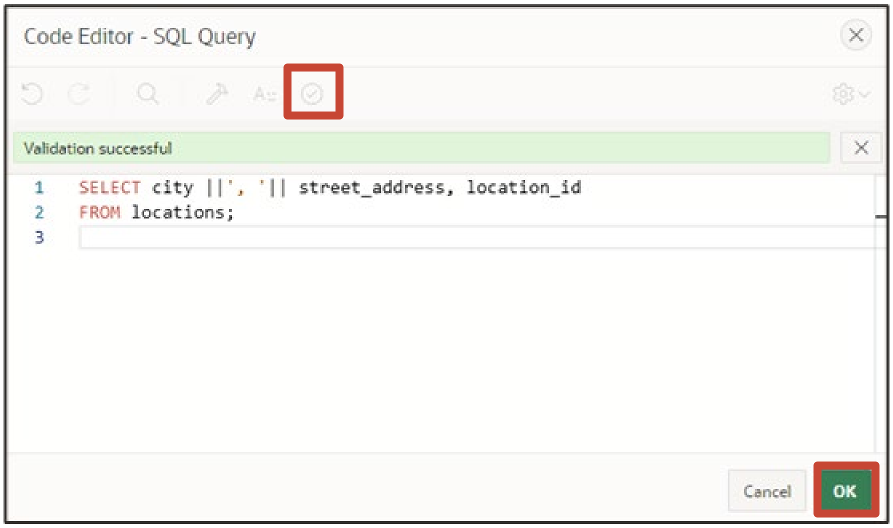


Value Required
Type: SQL Query
SQL Query
Display Extra Values
Display Null Value
Null Display Value
Null Return Value

3. Enter the following code in the code editor:

```
SELECT city ||', '|| street_address, location_id
FROM locations;
```

4. Click the **Validate** button. Once successful, click **OK**.



Validation successful

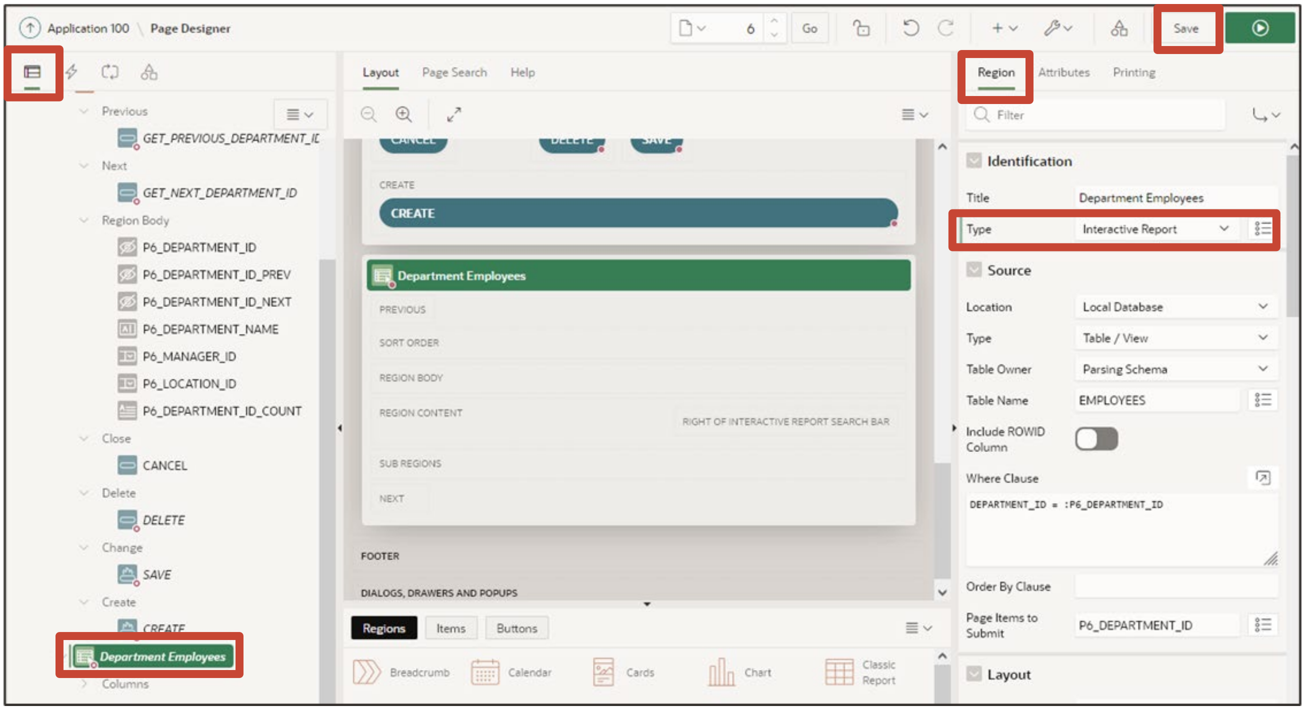
```
SELECT city ||', '|| street_address, location_id
FROM locations;
```

5. Click **Save** when returned to **Page Designer** view.

Step 7: Set Employee Details to be View Only

By default the employees are displayed in an editable interactive grid, which allows users to update employee records. We want users to manage employee records on the existing **Update Employee** page. Ensure **Page 6 – Department Employees** is open in **Page Designer**.

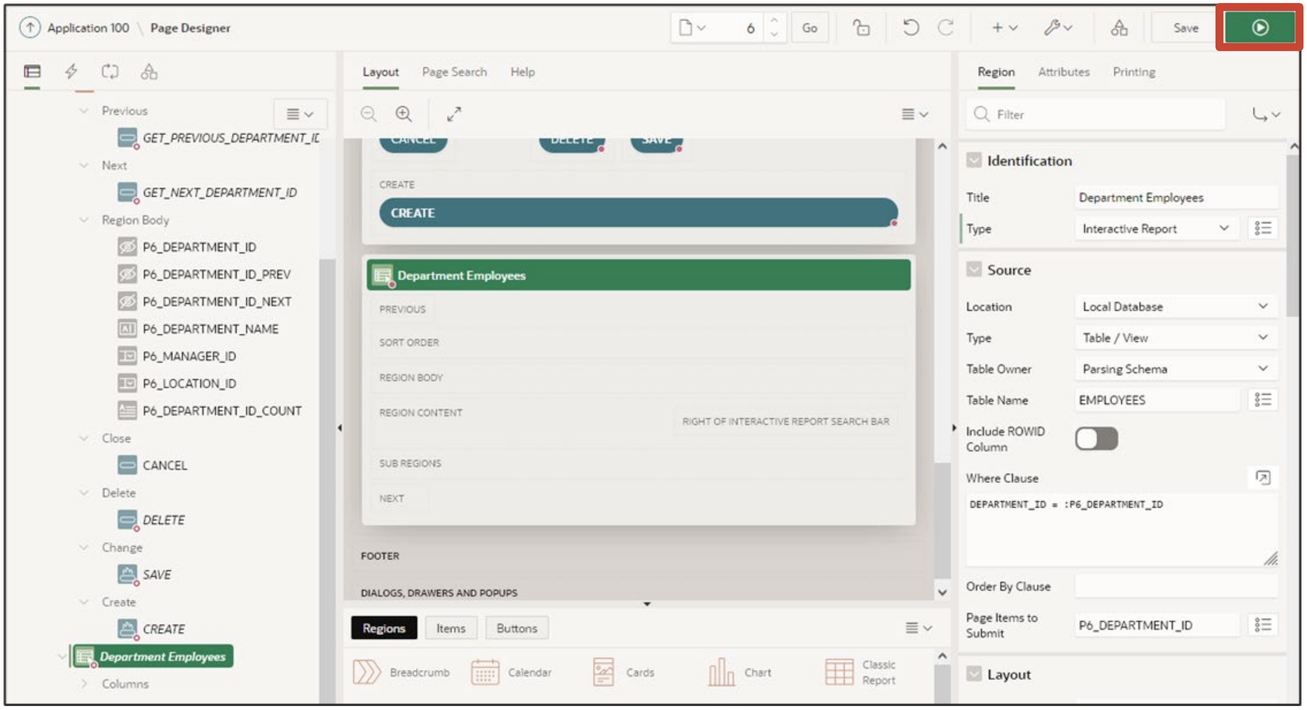
1. From the rendering tab, scroll down to **Department Employees** (it is near the bottom of the rendering tab) and left click to select the form. In the **Region** tab, change **Type** from **Interactive Grid** to **Interactive Report**. Click **Save**.



Title: Department Employees
Type: Interactive Report

Location: Local Database
Type: Table / View
Table Owner: Parsing Schema
Table Name: EMPLOYEES
Include ROWID Column
Where Clause: DEPARTMENT_ID = :P6_DEPARTMENT_ID
Order By Clause
Page Items to Submit: P6_DEPARTMENT_ID

2. **Run** your application to test. Do not create a new department at this time.



Title: Department Employees
Type: Interactive Report
Location: Local Database
Type: Table / View
Table Owner: Parsing Schema
Table Name: EMPLOYEES
Include ROWID Column
Where Clause:

```
DEPARTMENT_ID = :P6_DEPARTMENT_ID
```

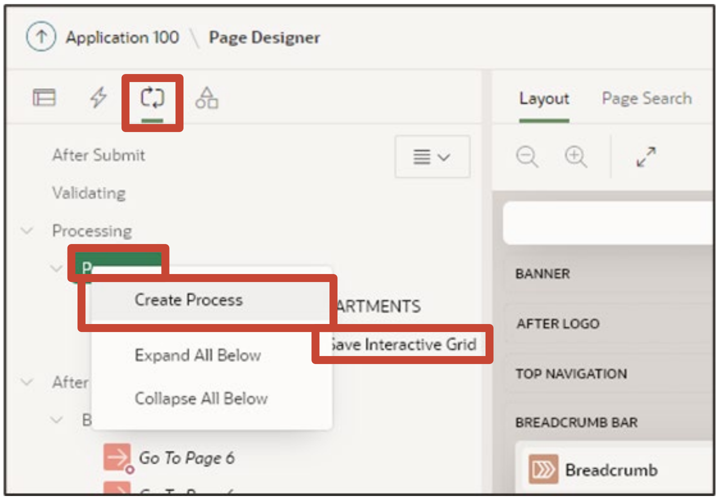
Order By Clause
Page Items to Submit: P6_DEPARTMENT_ID

Part 6: Use a Sequence to Manage Primary Key Values on the Update Departments Page

In this step, we will create a new process to automatically generate primary key values using an existing sequence when creating a new department the same way we did earlier for the `employee_id` primary key.

Step 1: Create a New Process

1. Right click on **Processes** and select **Create Process**.

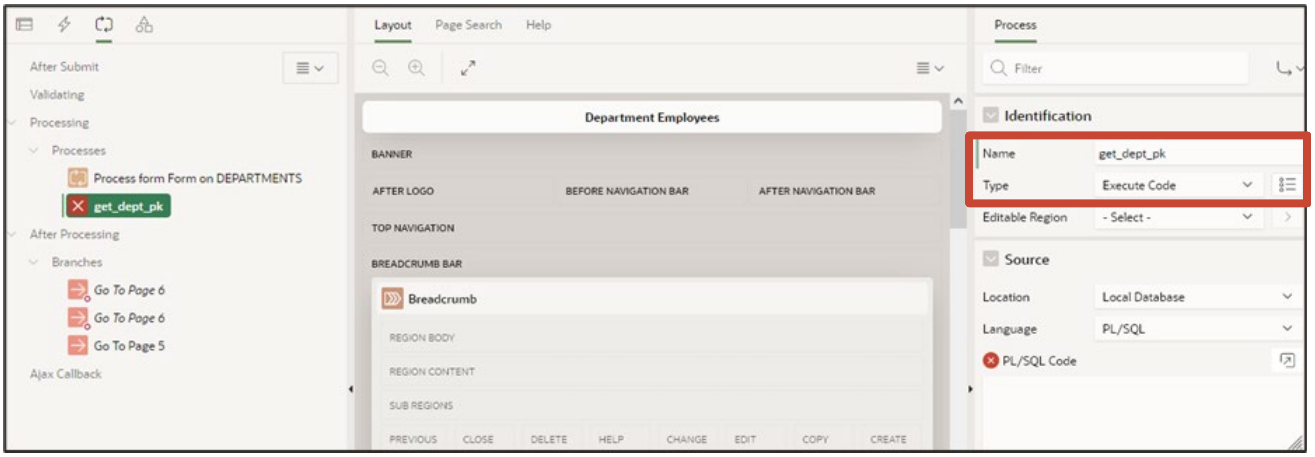


Step 2: Enter Code for the Process

1. Enter the following for **Identification**:

   Name: `get_dept_pk`

   Type: Execute Code

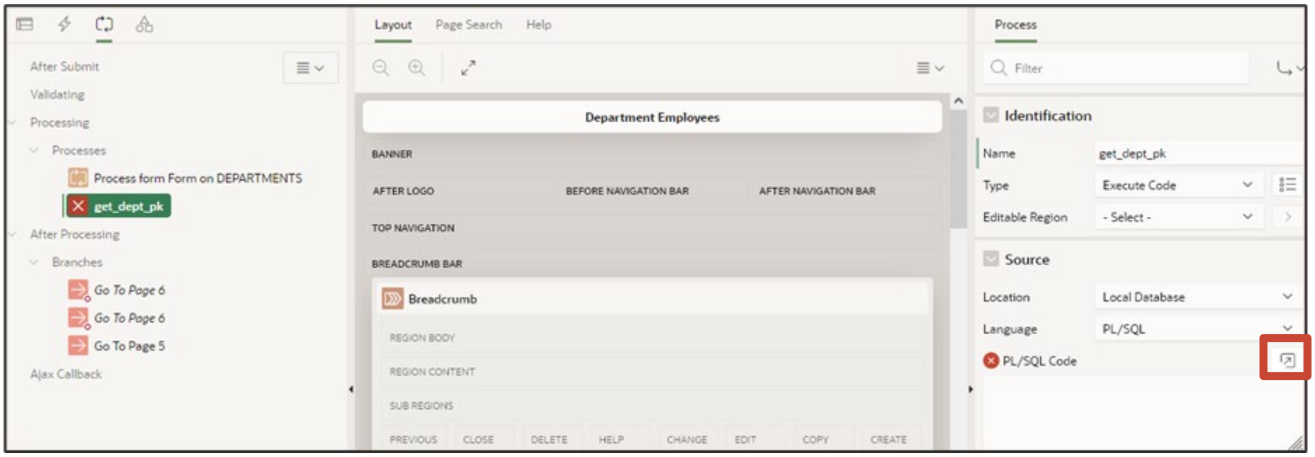


Processing > Processes: Process form Form on DEPARTMENTS, get_dept_pk

Identification: Name: get_dept_pk; Type: Execute Code; Editable Region: - Select -

Source: Location: Local Database; Language: PL/SQL; PL/SQL Code

2. Ensure the following for **Source** details. Then click the icon to open the code editor.



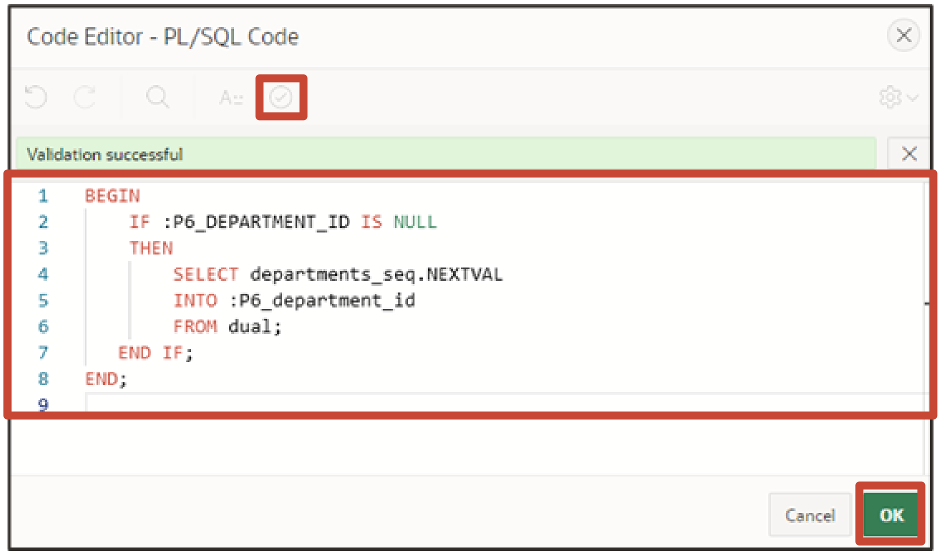
Identification: Name: get_dept_pk; Type: Execute Code; Editable Region: - Select -

Source: Location: Local Database; Language: PL/SQL; PL/SQL Code

3. Enter the following code in the code editor:

```
BEGIN
    IF :P6_DEPARTMENT_ID IS NULL
    THEN
        SELECT departments_seq.NEXTVAL
        INTO :P6_department_id
        FROM dual;
   END IF;
END;
```

4. Once the statement is entered, click the **Validation** icon. Once successful, click **OK**.

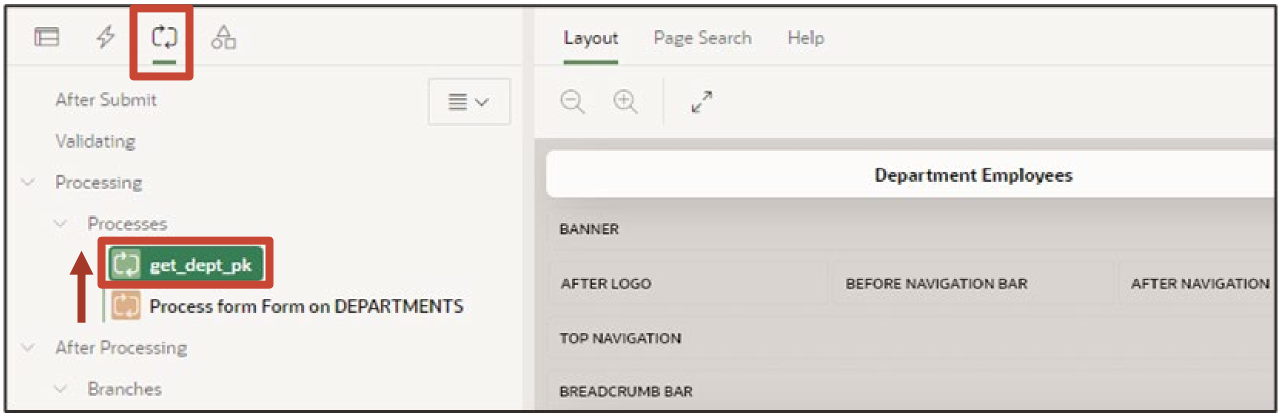


Validation successful

```
BEGIN
  IF :P6_DEPARTMENT_ID IS NULL
  THEN
    SELECT departments_seq.NEXTVAL
    INTO :P6_department_id
    FROM dual;
  END IF;
END;
```

5. The new **`get_dept_pk`** process must run before the other process to ensure that `department_id` has a value before the insert is processed.

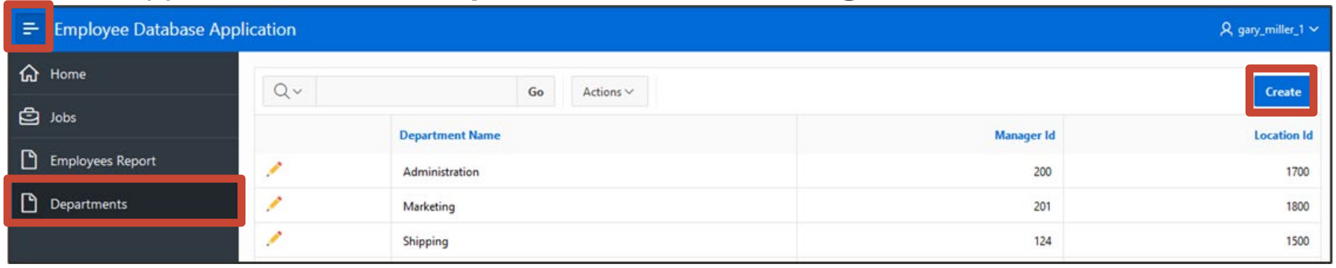
   In the processes tab, left mouse click and drag the new **`get_dept_pk`** process so that it is above the **Process form Form on DEPARTMENTS**.



6. **Save** the page.

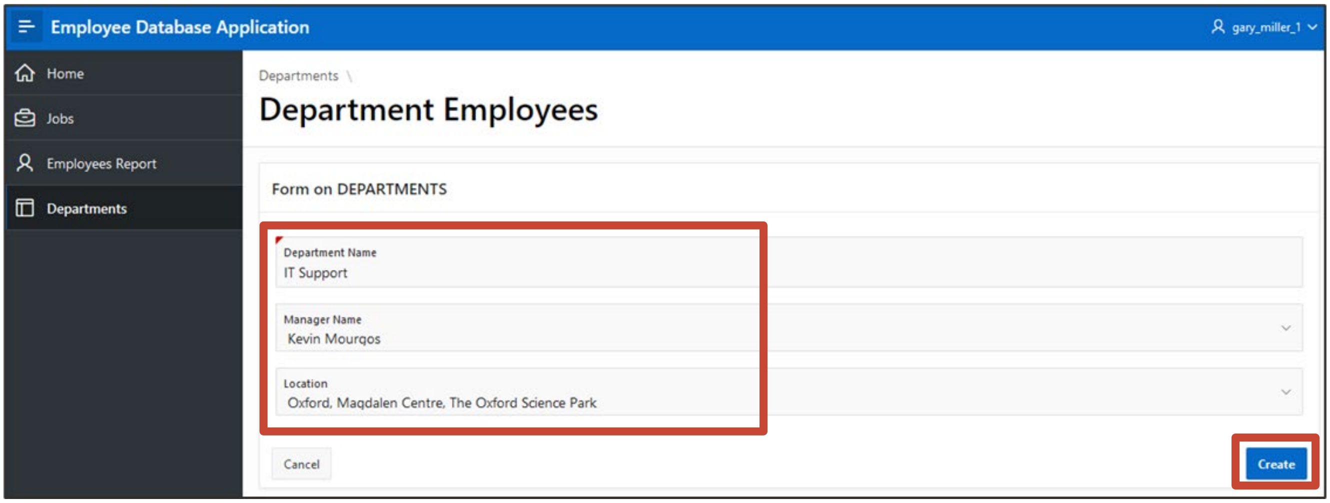
Step 3: Run and Test

1. Run the application and click **Departments** from the **Navigation Menu**, then click **Create**.



Department Name	Manager Id	Location Id
Administration	200	1700
Marketing	201	1800
Shipping	124	1500

2. Enter values for a new department, then click **Create**.

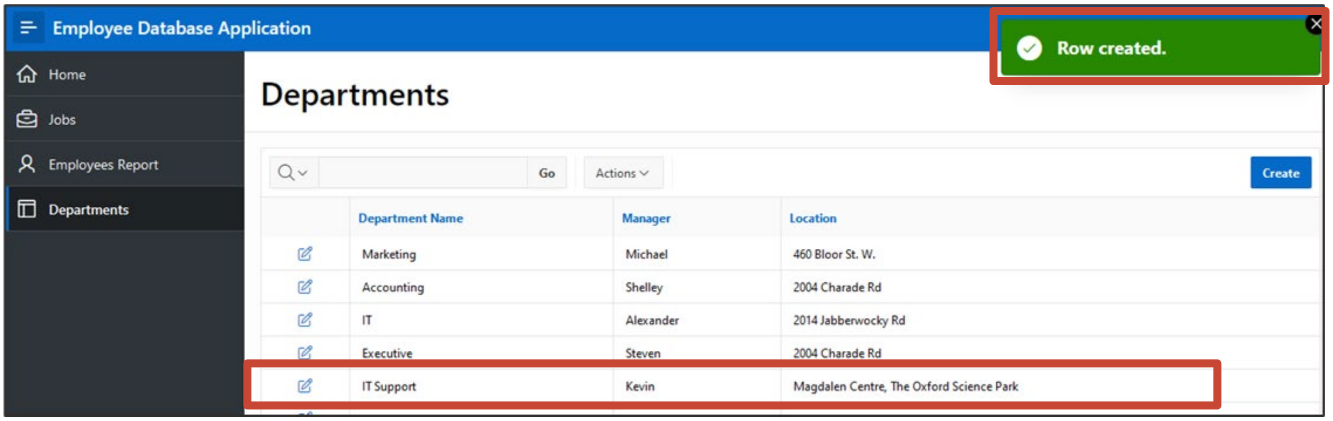


Department Name: IT Support

Manager Name: Kevin Mourgos

Location: Oxford, Magdalen Centre, The Oxford Science Park

3. The new department should be displayed in the Department Report, and a **Row created** message will be displayed.



Row created.

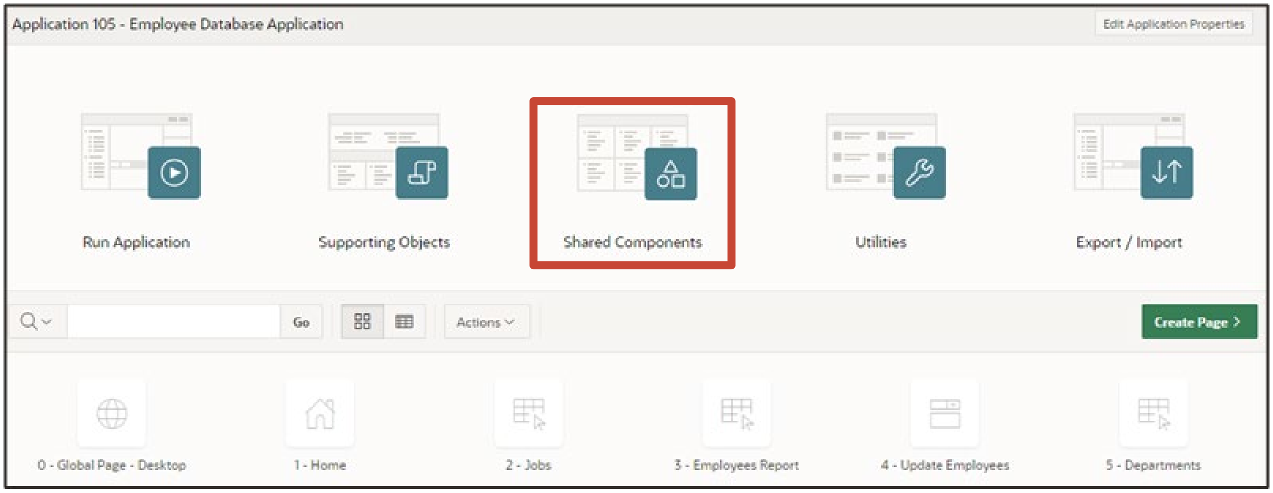
Department Name	Manager	Location
Marketing	Michael	460 Bloor St. W.
Accounting	Shelley	2004 Charade Rd
IT	Alexander	2014 Jabberwocky Rd
Executive	Steven	2004 Charade Rd
IT Support	Kevin	Magdalen Centre, The Oxford Science Park

Part 7: Improve Appearance of Home Page and Navigation Menu

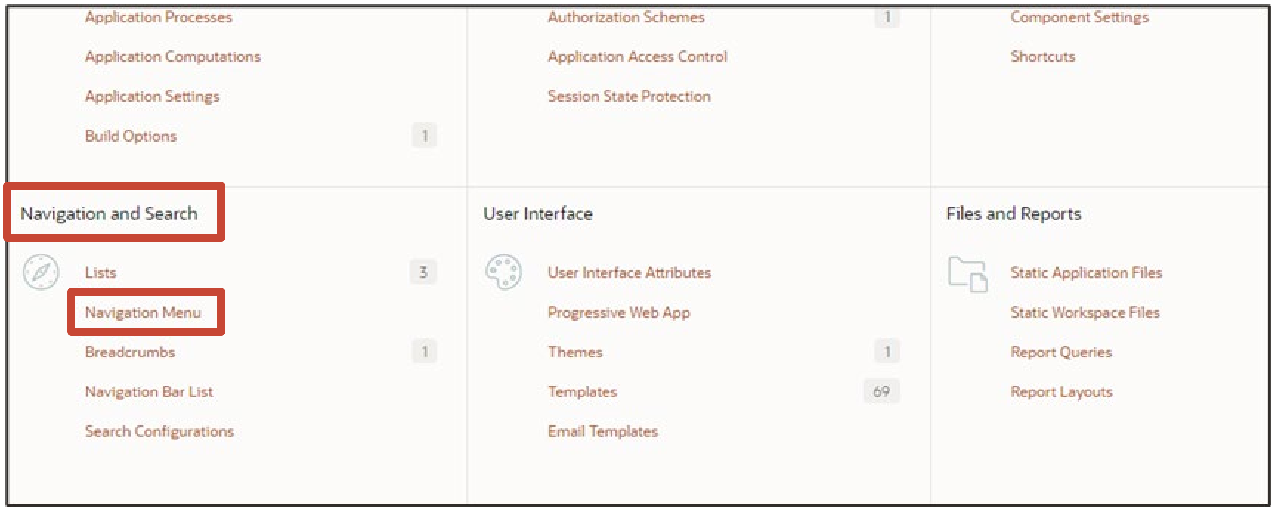
You can change the menu item icon to better match the menu item purpose. In the example below, the default icon was accepted when the page was created. In this section, we will update **Navigation Menu** icons, and add tiles to the home page for the other pages also.

Step 1: Change the Navigation Menu Icons

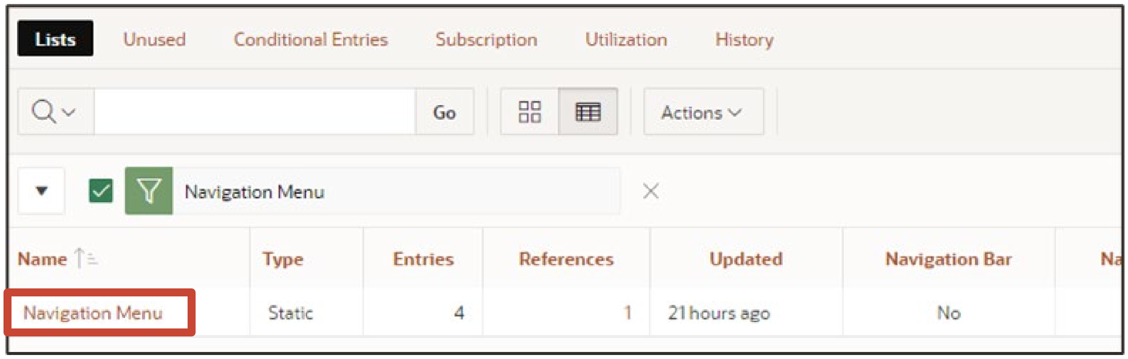
1. From the **Employee Database Application** home page, click **Shared Components**.



2. In the **Navigation and Search** section, click **Navigation Menu**.

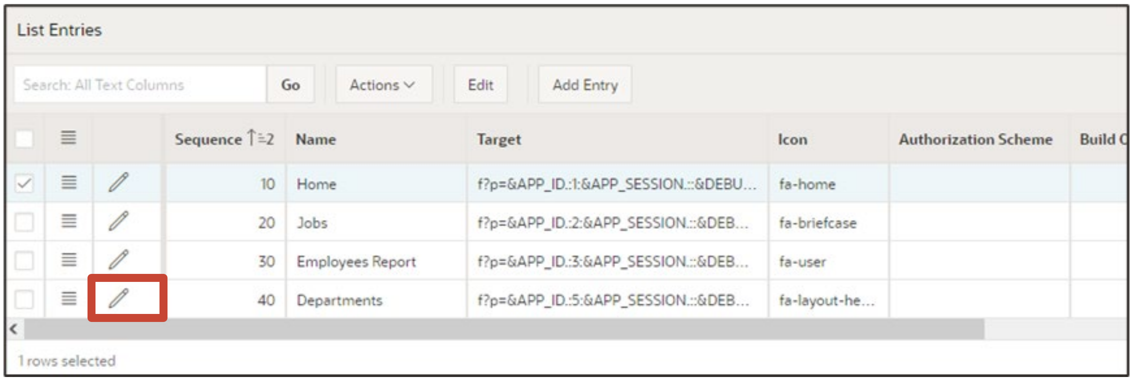


3. Click **Navigation Menu**.



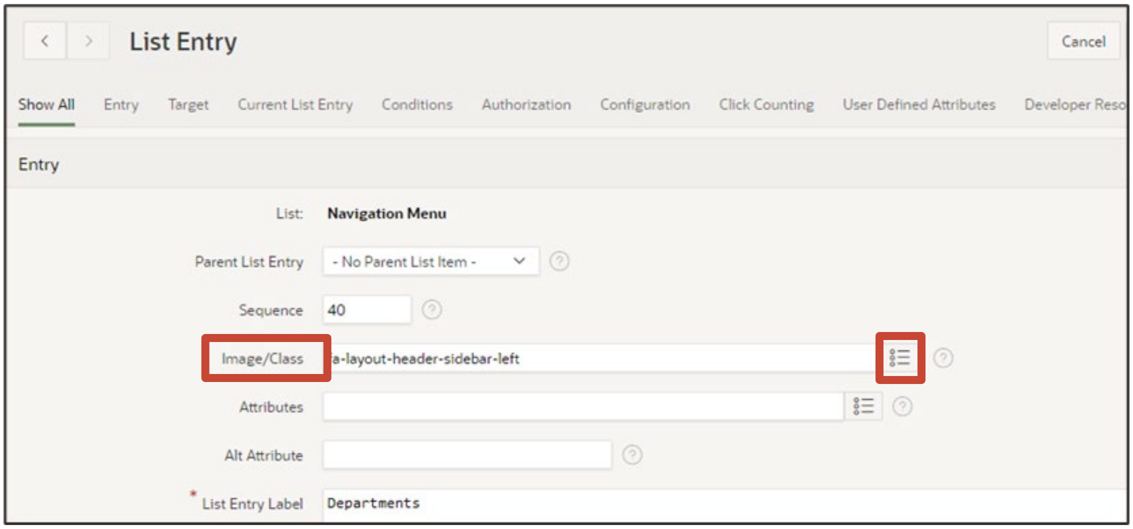
Name	Type	Entries	References	Updated	Navigation Bar	Na
Navigation Menu	Static	4	1	21 hours ago	No	

4. Click the **Edit** icon for **Departments**.



☐			Sequence ↑	Name	Target	Icon	Authorization Scheme	Build C
☑	≡	✎	10	Home	f?p=&APP_ID.:1:&APP_SESSION.::&DEBU...	fa-home		
☐	≡	✎	20	Jobs	f?p=&APP_ID.:2:&APP_SESSION.::&DEB...	fa-briefcase		
☐	≡	✎	30	Employees Report	f?p=&APP_ID.:3:&APP_SESSION.::&DEB...	fa-user		
☐	≡	✎	40	Departments	f?p=&APP_ID.:5:&APP_SESSION.::&DEB...	fa-layout-he...		

5. For **Image/Class**, click the **List** icon.



Entry

List: **Navigation Menu**

Parent List Entry: - No Parent List Item -

Sequence: 40

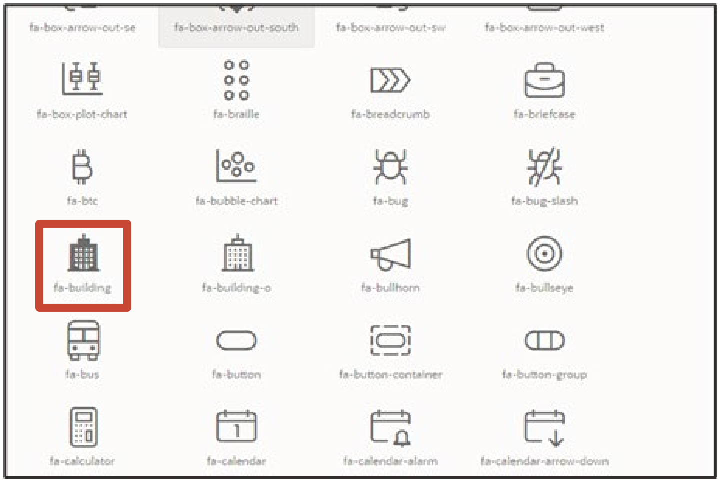
Image/Class: a-layout-header-sidebar-left

Attributes:

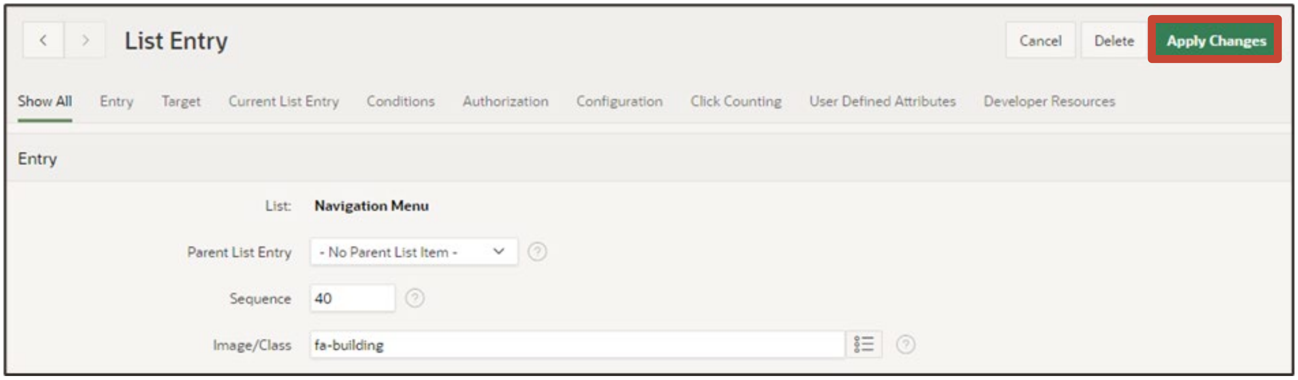
Alt Attribute:

* List Entry Label: Departments

6. Scroll down through the icons and left click the **fa-building** icon.



7. Click **Apply Changes**.



List: Navigation Menu

Parent List Entry: - No Parent List Item -

Sequence: 40

Image/Class: fa-building

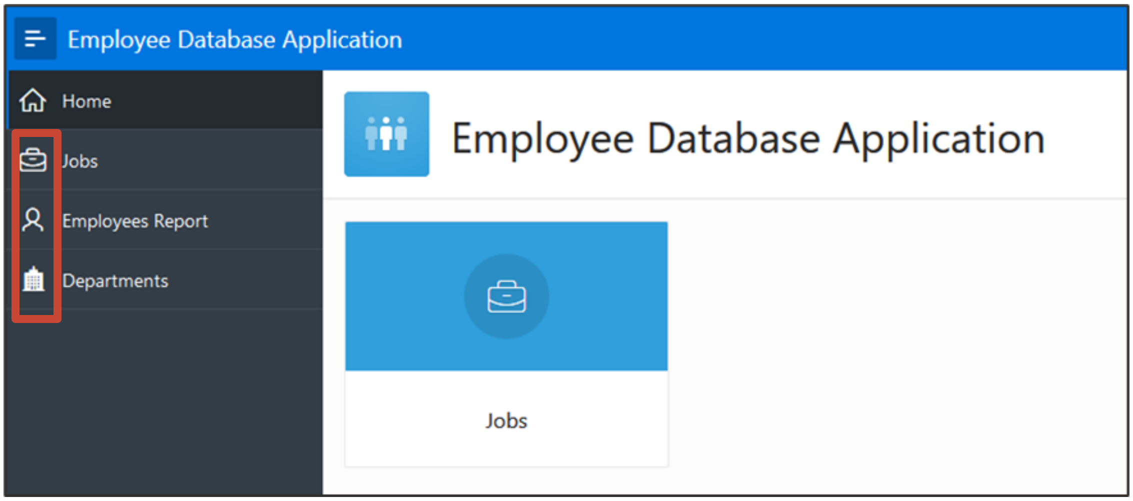
8. Follow the same process to change any other icons as required. The example below uses:

   Jobs: fa-briefcase

   Employee Report: fa-user

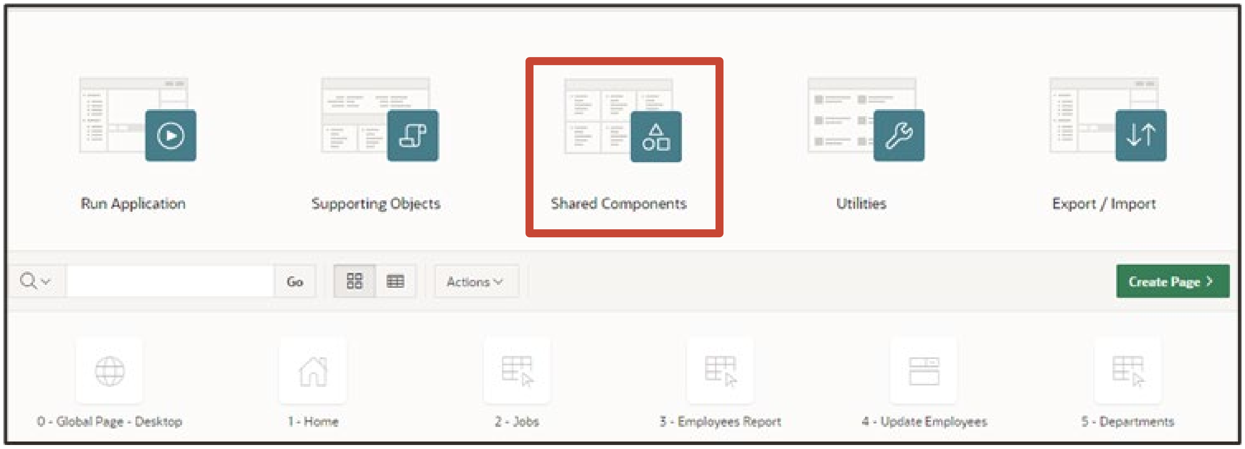
   Departments: fa-building-o

   Run the application to test. The new icons will now be shown in the **Navigation Menu**.

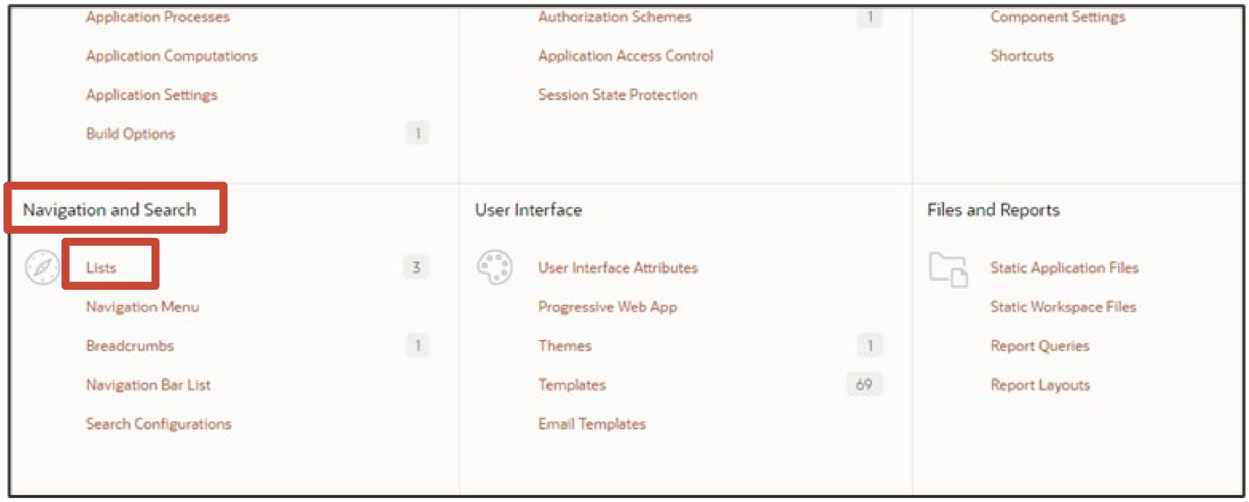


Step 2: Add Navigation Tiles to the Home Page

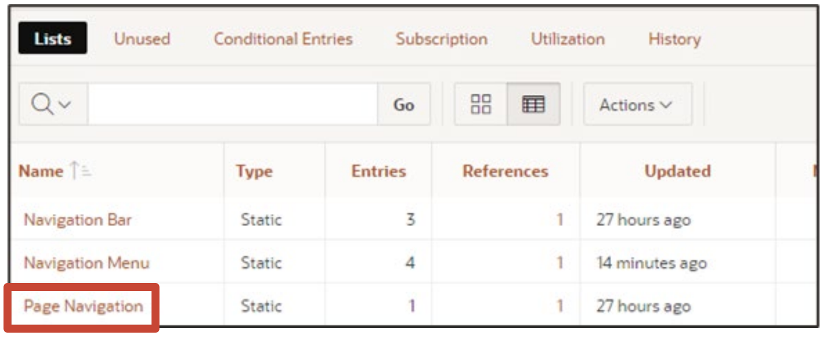
1. From the **Employee Database Application** home page, click **Shared Components**.



2. In the **Navigation and Search** section, click **Lists**.

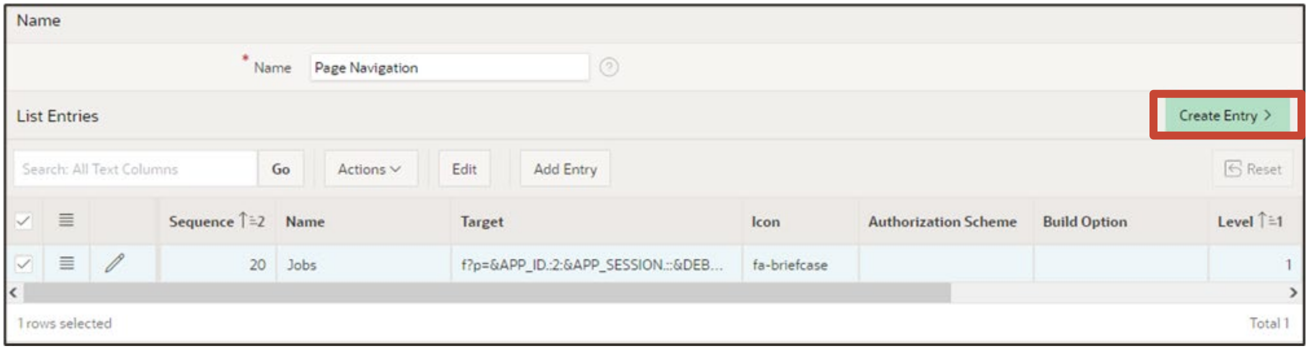


3. Click **Page Navigation**.



Name	Type	Entries	References	Updated
Navigation Bar	Static	3	1	27 hours ago
Navigation Menu	Static	4	1	14 minutes ago
Page Navigation	Static	1	1	27 hours ago

4. Click **Create Entry**.



Name

* Name: Page Navigation

List Entries

			Sequence ↑=2	Name	Target	Icon	Authorization Scheme	Build Option	Level ↑=1
☑	≡	✎	20	Jobs	f?p=&APP_ID.:2:&APP_SESSION.::&DEB...	fa-briefcase			1

1 rows selected — Total 1

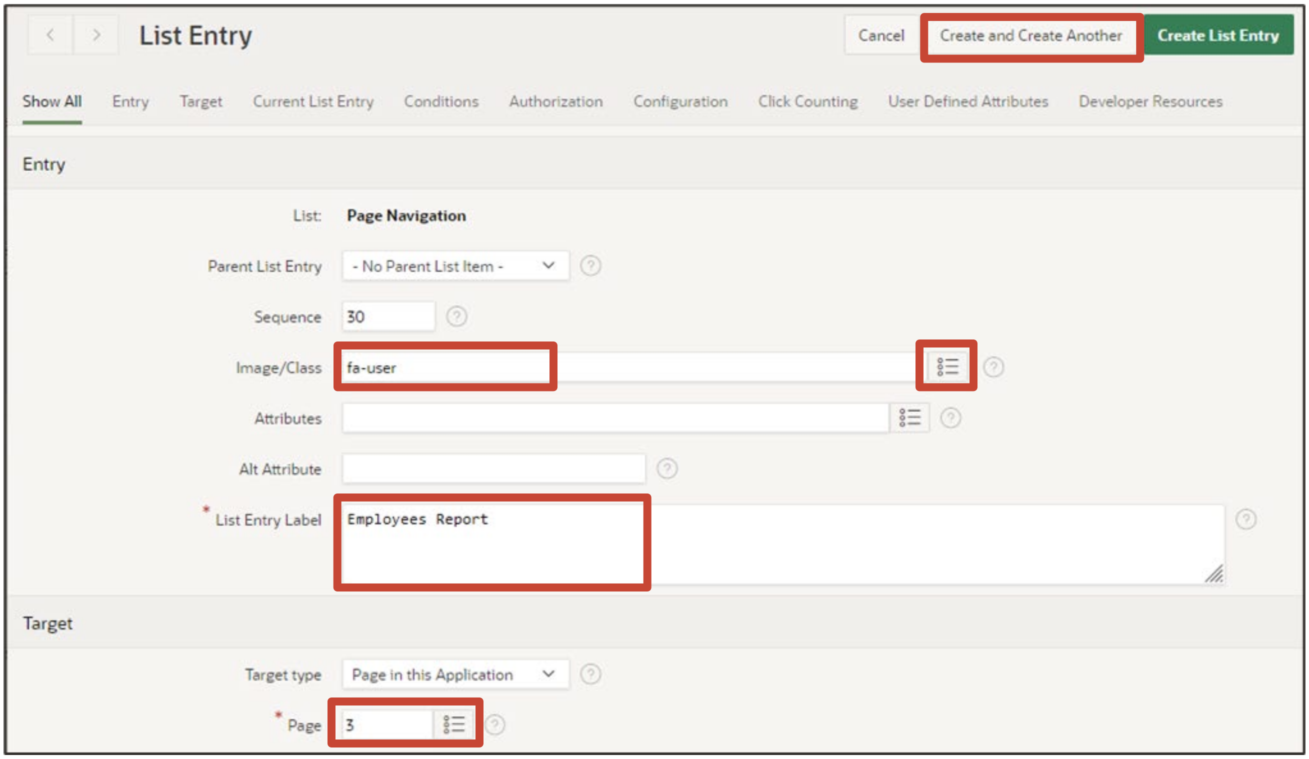
5. Enter the following properties:

Image/Class: fa-user

List Entry Label: Employees Report

 Page: 3

Click **Create and Create Another**.



List Entry

Entry

List: **Page Navigation**

Parent List Entry: - No Parent List Item -

Sequence: 30

Image/Class: fa-user

Attributes:

Alt Attribute:

* List Entry Label: Employees Report

Target

Target type: Page in this Application

* Page: 3

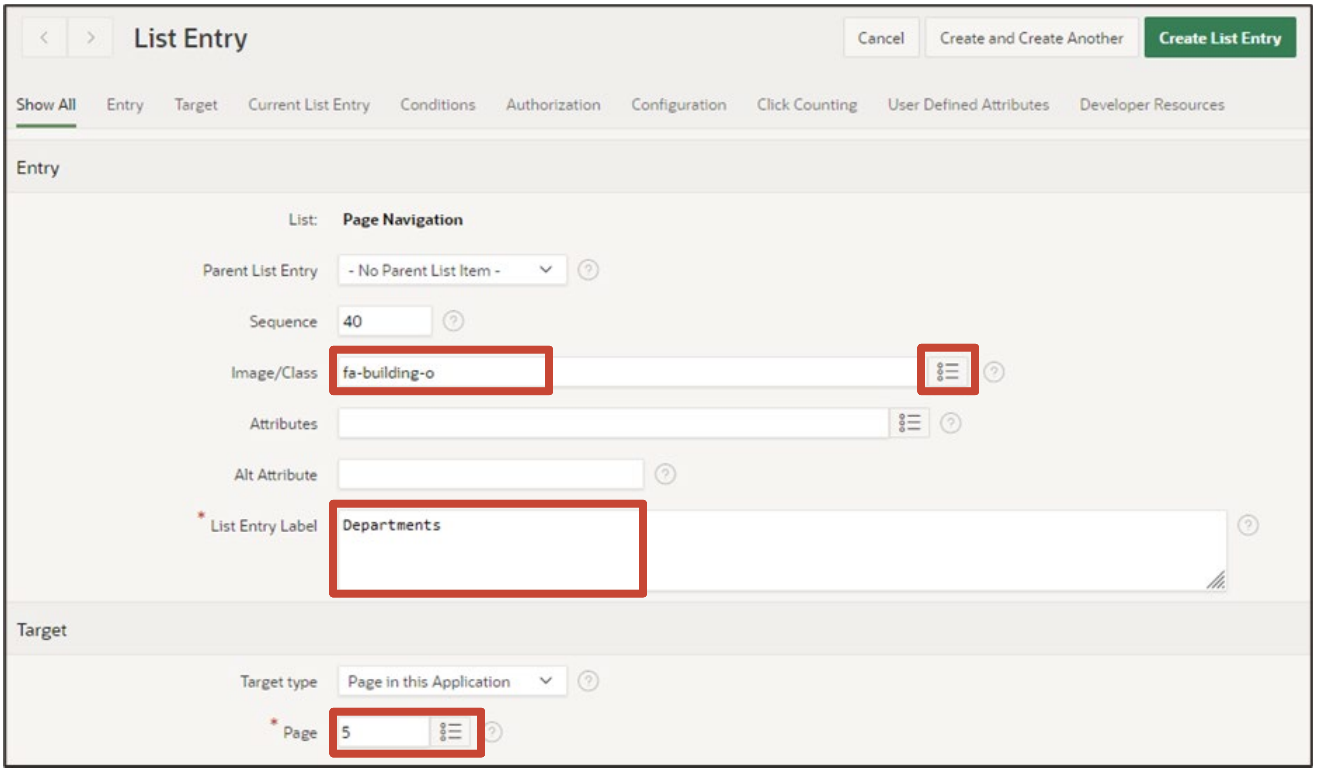
6. Enter the following properties:

   Image/Class: fa-building-o

   List Entry Label: Departments

   Page: 5

   Click **Create List Entry**



List Entry

Entry

List: **Page Navigation**

Parent List Entry: - No Parent List Item -

Sequence: 40

Image/Class: fa-building-o

Attributes:

Alt Attribute:

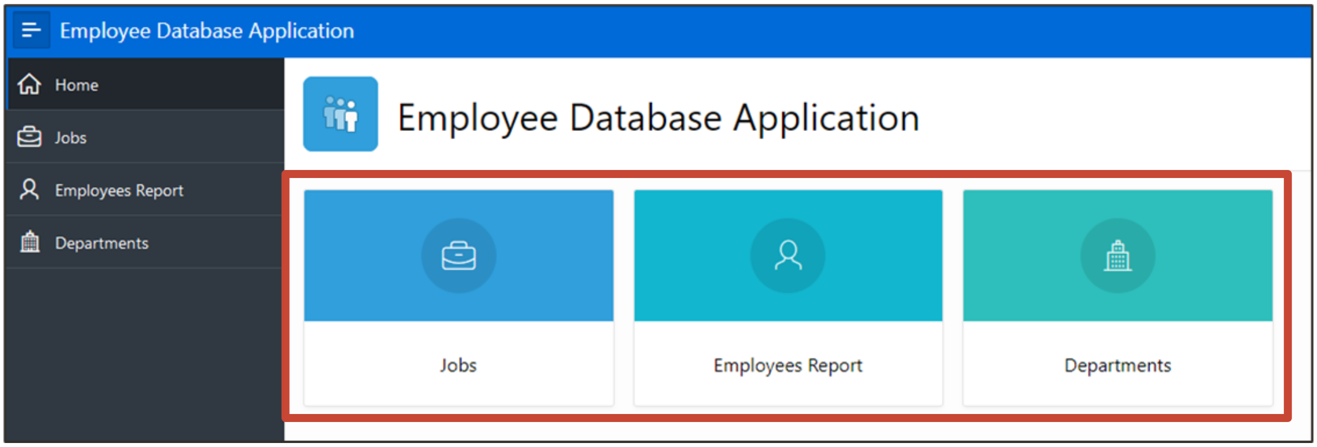
* List Entry Label: Departments

Target

Target type: Page in this Application

* Page: 5

7. Return to the **Employee Database Application** home page and run the application to test. There should now be 3 navigation tiles showing.

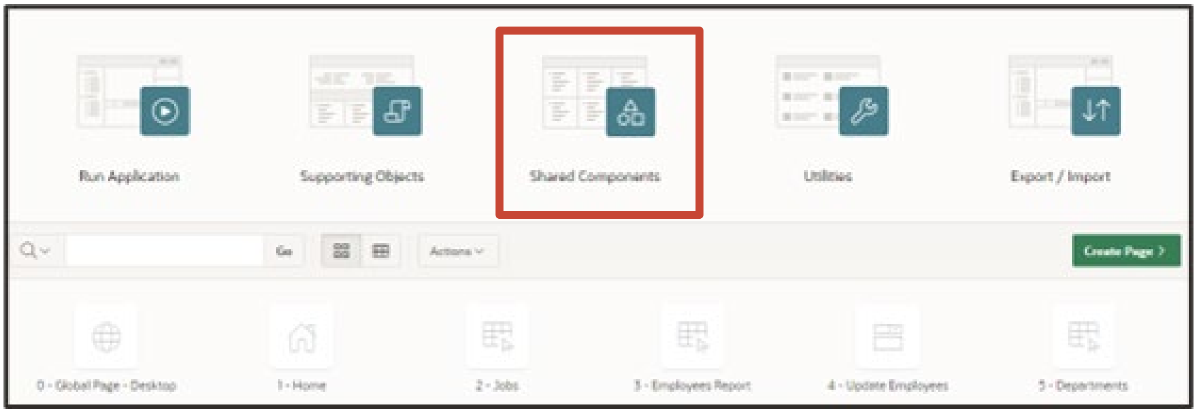


Part 8: Add an Image to the Home Page

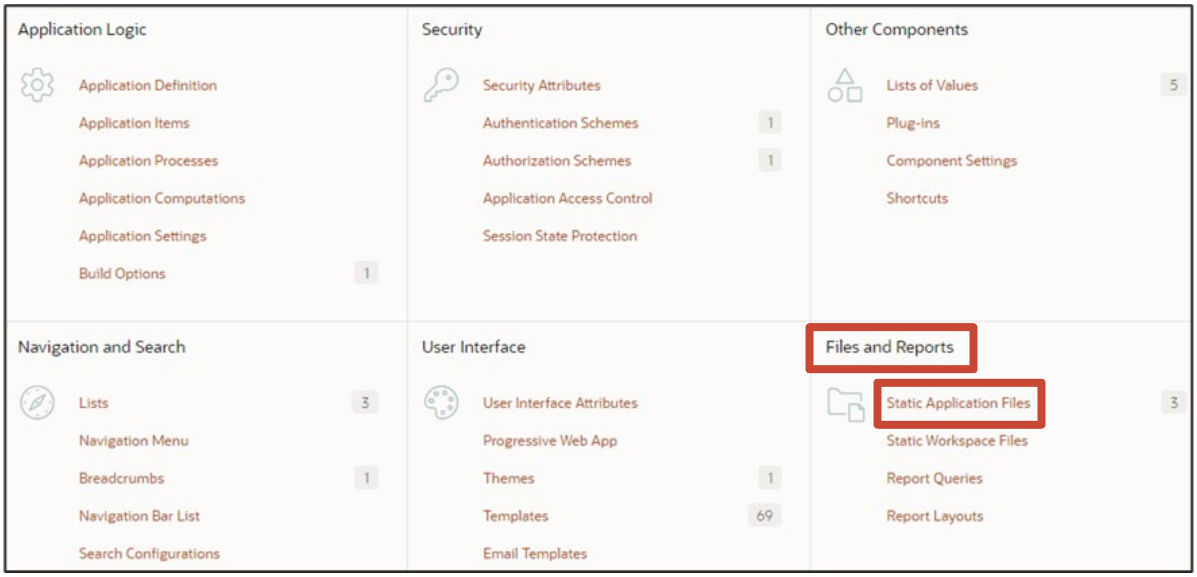
First find an image you want to use on the home page, and save locally on your PC, noting its location. Make sure the image is not too large, as we want it to fit on screen without the user having to scroll across the page. The size should be no more than 450 pixels wide, and 200 pixels high.

Step 1: Upload the Image

1. From the **Employee Database Application** home page, click the **Shared Components** icon.



2. In the **Files and Reports** section, click **Static Application Files**.



Application Logic
Application Definition
Application Items
Application Processes
Application Computations
Application Settings
Build Options 1

Security
Security Attributes
Authentication Schemes 1
Authorization Schemes 1
Application Access Control
Session State Protection

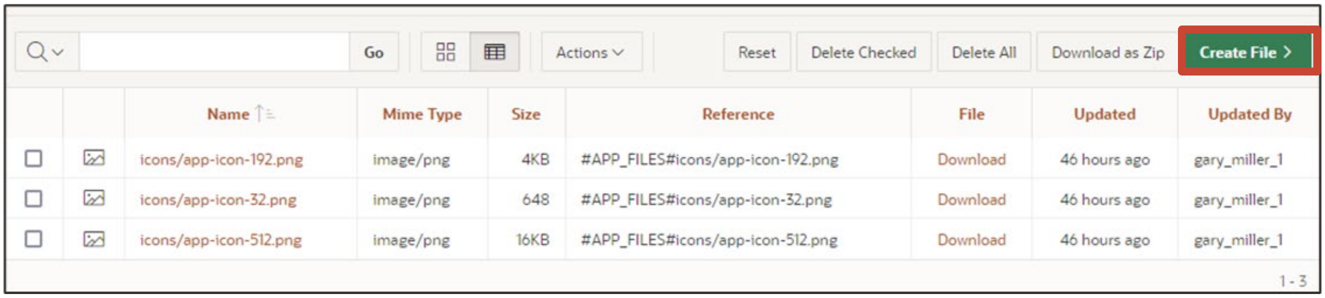
Other Components
Lists of Values 5
Plug-ins
Component Settings
Shortcuts

Navigation and Search
Lists 3
Navigation Menu
Breadcrumbs 1
Navigation Bar List
Search Configurations

User Interface
User Interface Attributes
Progressive Web App
Themes 1
Templates 69
Email Templates

Files and Reports
Static Application Files 3
Static Workspace Files
Report Queries
Report Layouts

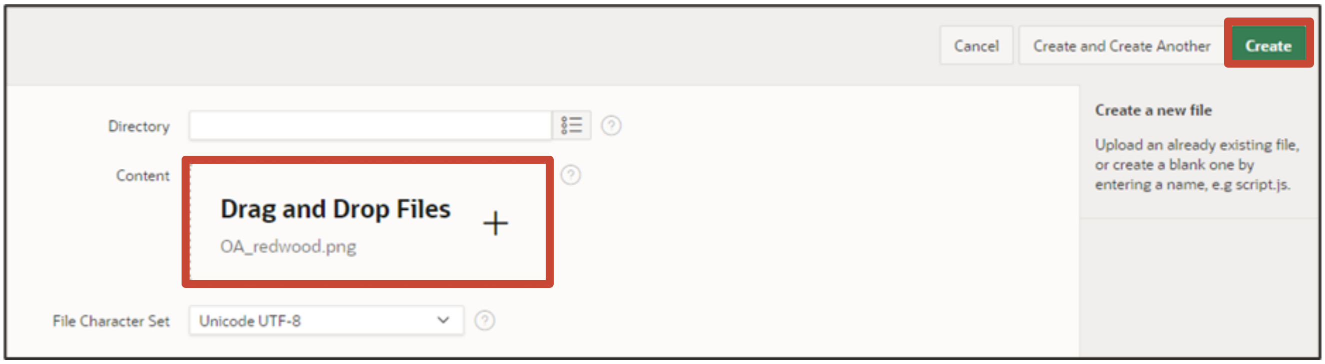
3. Click **Create File**.



		Name	Mime Type	Size	Reference	File	Updated	Updated By
□		icons/app-icon-192.png	image/png	4KB	#APP_FILES#icons/app-icon-192.png	Download	46 hours ago	gary_miller_1
□		icons/app-icon-32.png	image/png	648	#APP_FILES#icons/app-icon-32.png	Download	46 hours ago	gary_miller_1
□		icons/app-icon-512.png	image/png	16KB	#APP_FILES#icons/app-icon-512.png	Download	46 hours ago	gary_miller_1

1 - 3

4. Locate the image you want to use on your local machine and drag and drop it into the **Content** area. Click **Create**.



Directory:

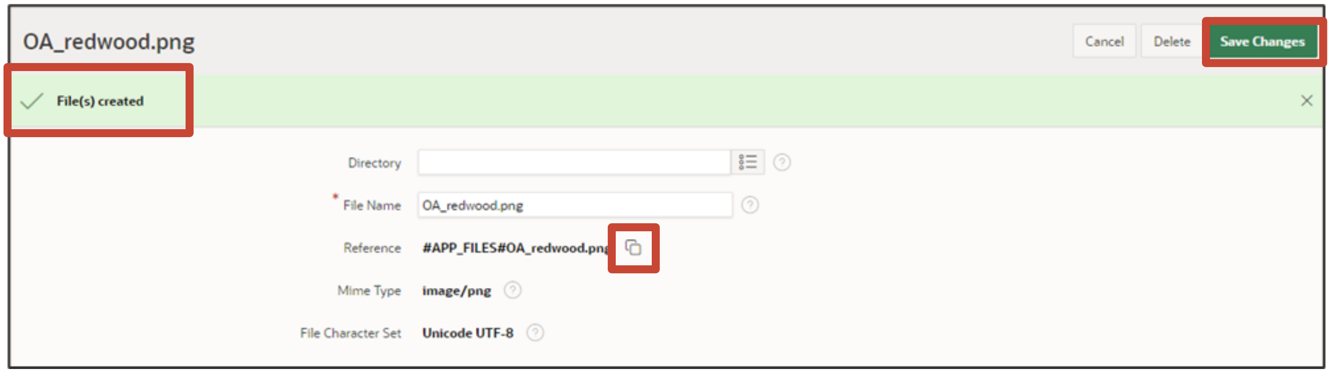
Content: OA_redwood.png

File Character Set: Unicode UTF-8

Create a new file

Upload an already existing file, or create a blank one by entering a name, e.g script.js.

5. Your file will be created. Copy the reference into a text document (click the **Copy** icon) as this is how your application will locate the file from the home page. Click **Save Changes**.



OA_redwood.png

✓ File(s) created

Directory:

* File Name: OA_redwood.png

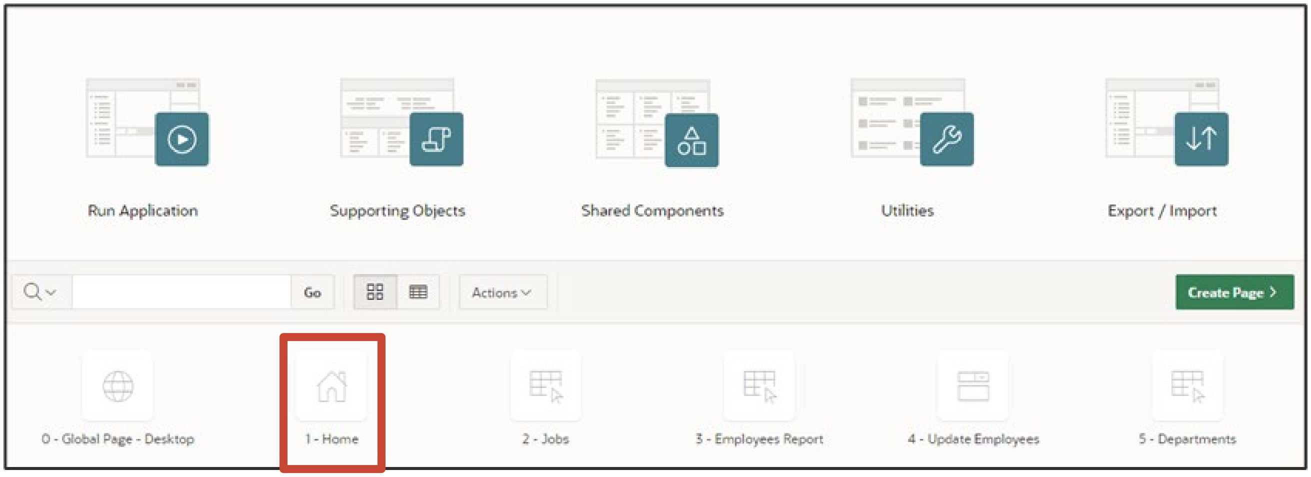
Reference: #APP_FILES#OA_redwood.png

Mime Type: image/png

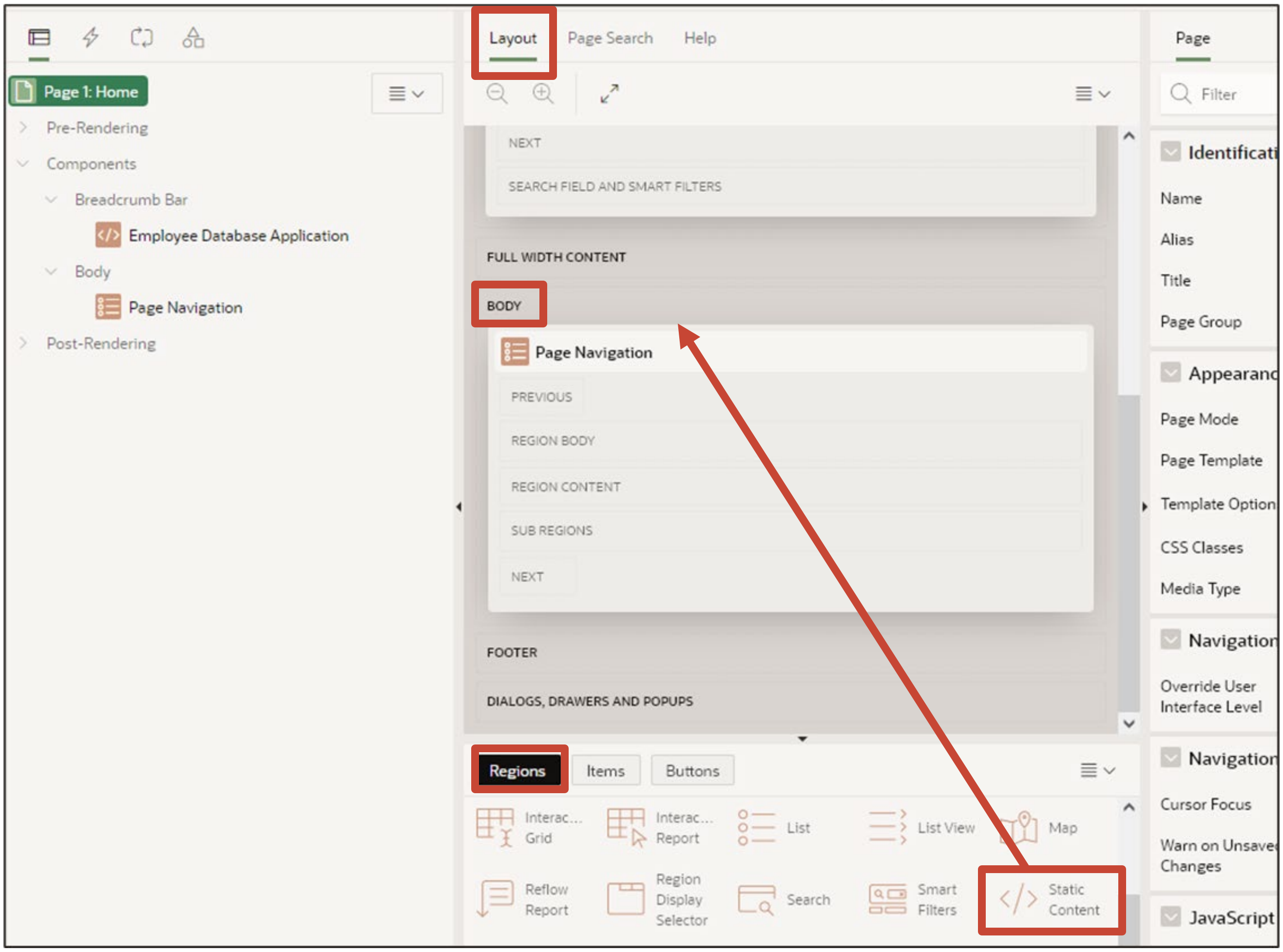
File Character Set: Unicode UTF-8

Step 2: Add a Region to the Home Page Content Body

1. Navigate back to the **Employee Database Application** home page, click page **1 – Home**.



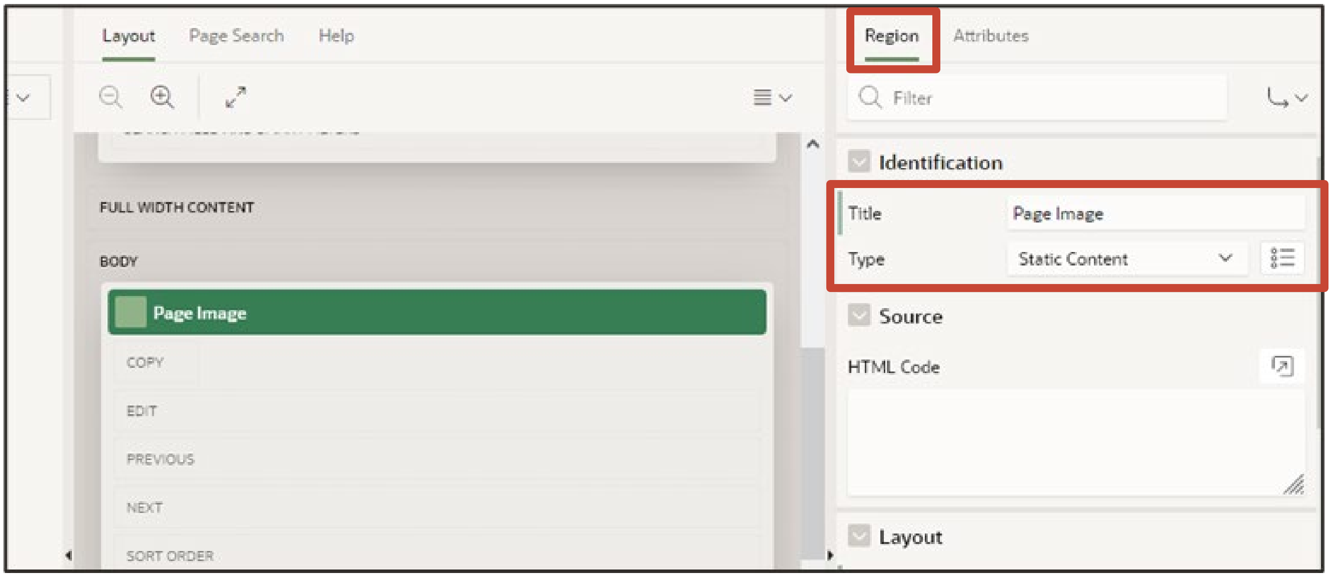
2. In the **Layout** section, scroll down to locate the **BODY** section. Locate and click the **Regions** tab. In the **Regions** section, scroll down to see the **Static Content** region. Drag the **Static Content** region into the content **BODY** above the existing entry for **Page Navigation**.



3. The new region will be shown directly below the **BODY** section. In the **Region** tab, enter the following properties:

   Title: Page Image

   Type: Static Content



Title: Page Image

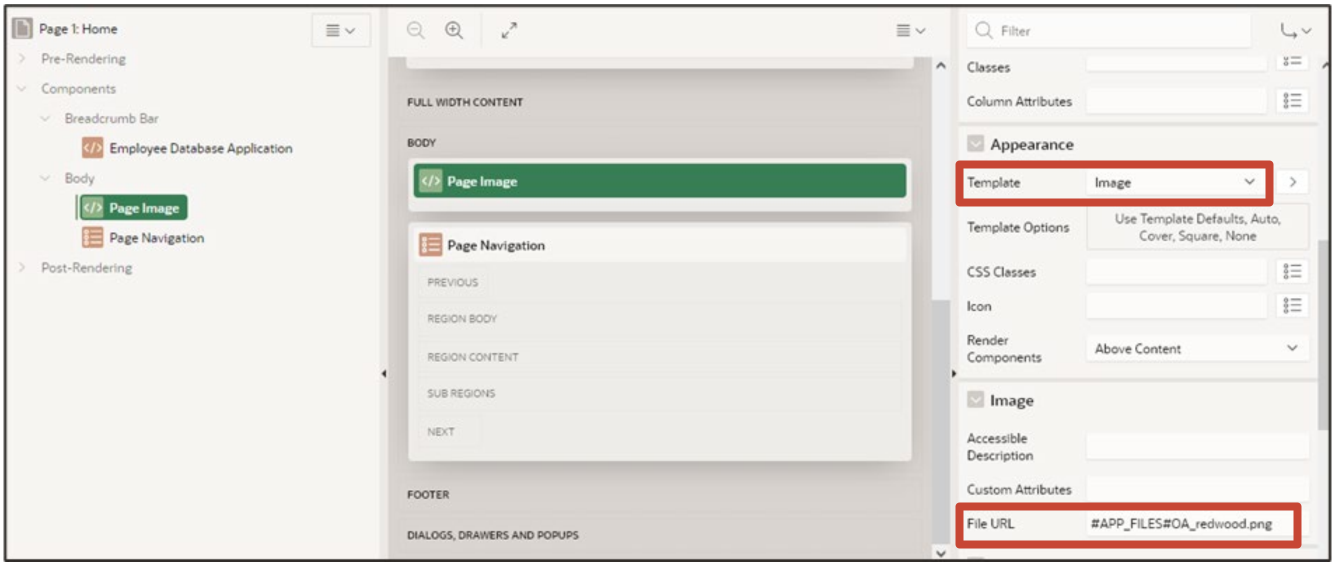
Type: Static Content

4. Still in the **Region** tab, scroll down to the **Appearance** section. Ensure the following properties:

   Template: Image

   File URL: Copy the file reference from Part 1

   To clear the error, **save** the page.



Template: Image

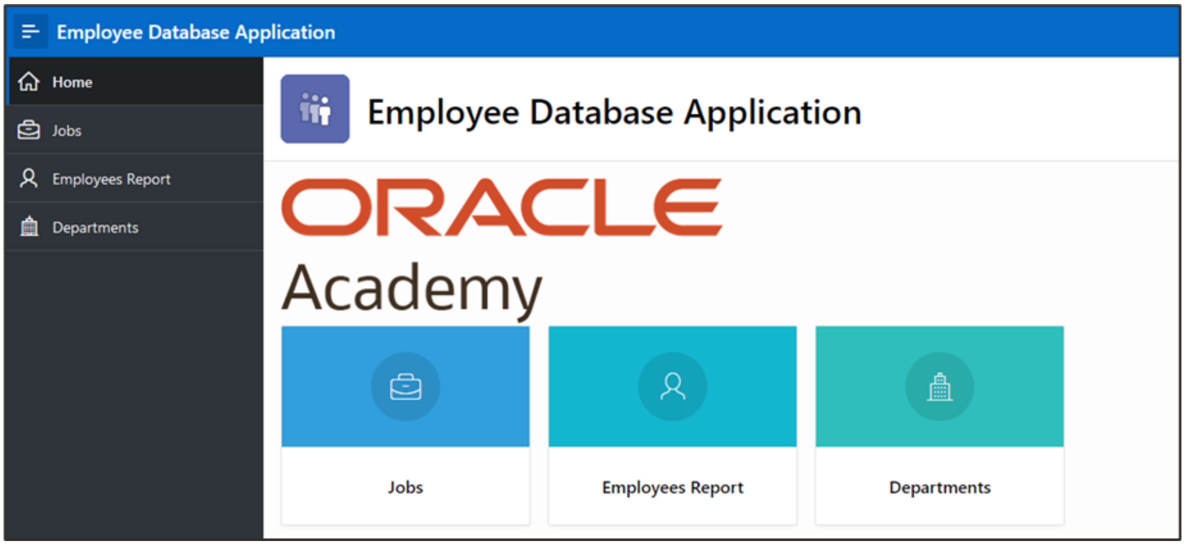
Template Options: Use Template Defaults, Auto, Cover, Square, None

Render Components: Above Content

File URL: #APP_FILES#OA_redwood.png

Step 3: Test the Application

1. Run the application to test. Your home page should now display your image, similar to the example below.



Employee Database Application
Home
Jobs
Employees Report
Departments
Employee Database Application
ORACLE
Academy
Jobs
Employees Report
Departments