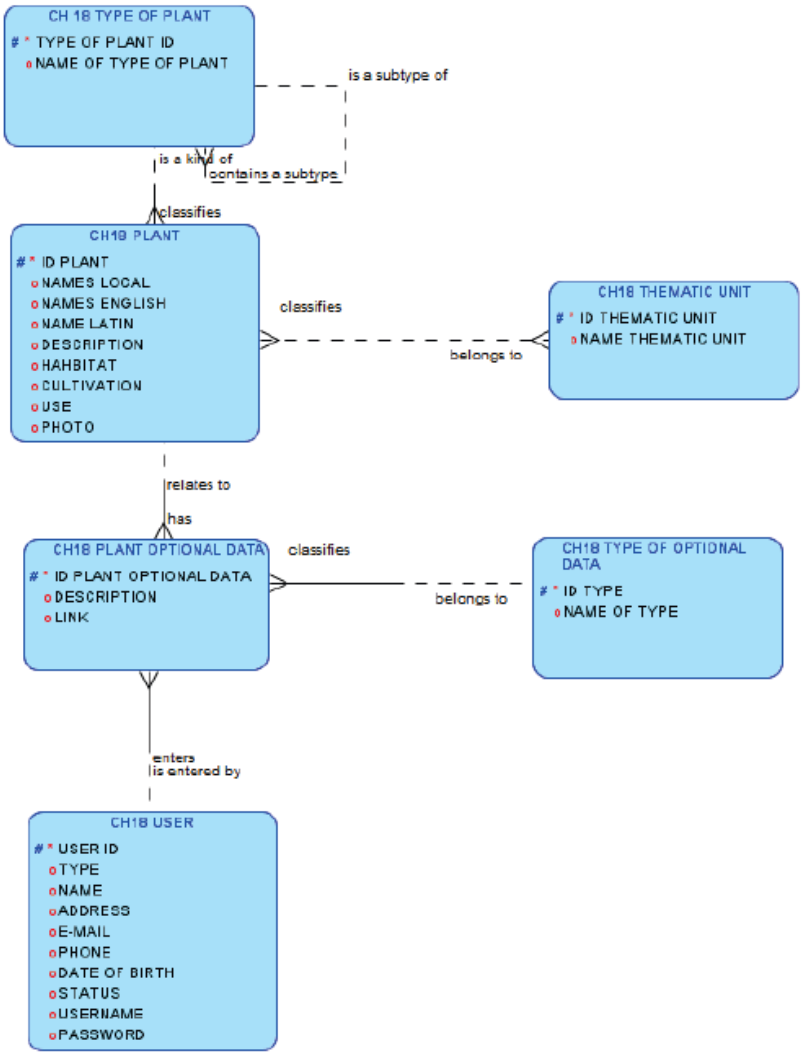
14. GreenDi - Catalog of plants

Chapter 14 from the book.

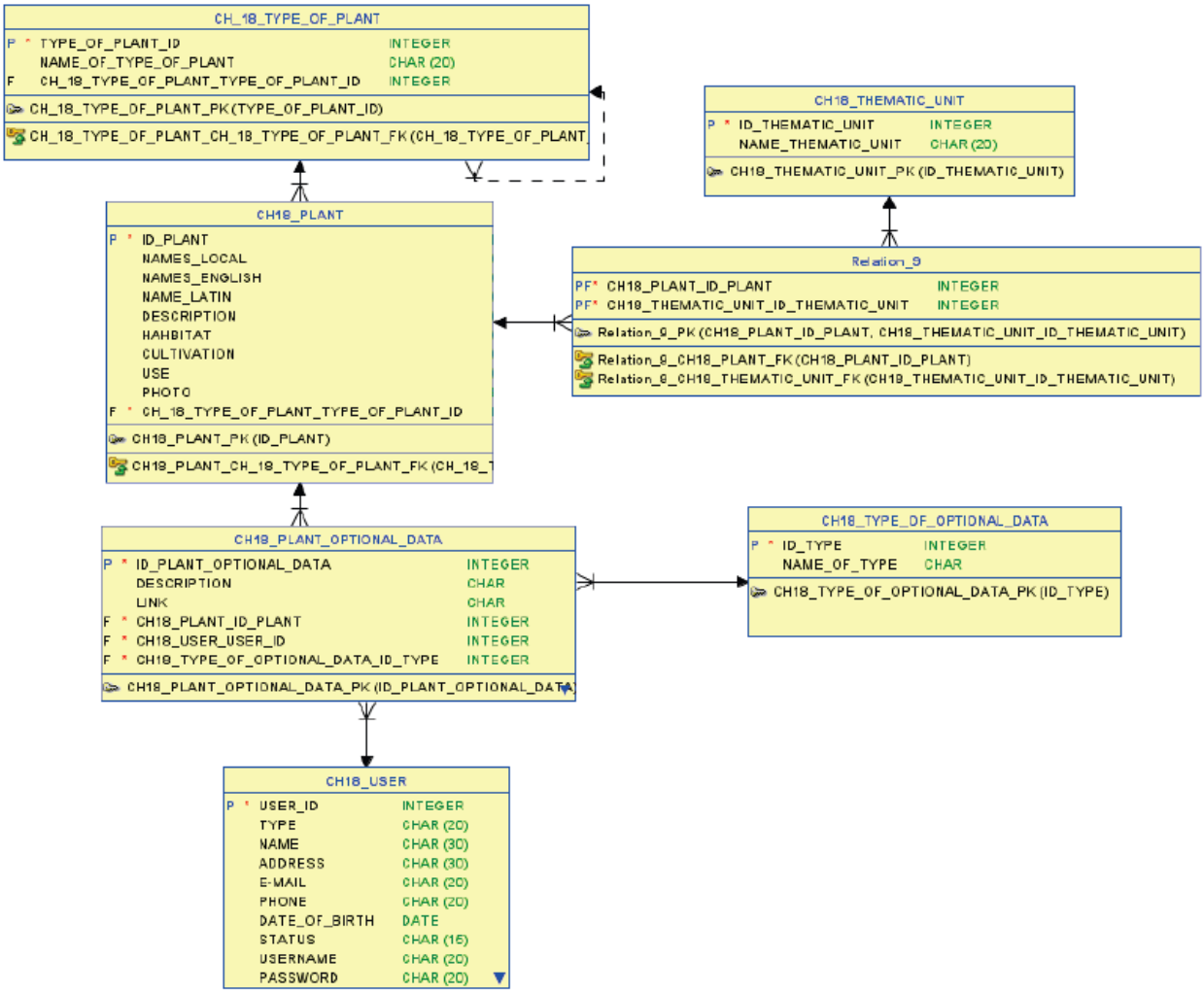
Create tables in database

Oracle Apex is database based low code platform, so the first step is to create database (tables, primary and foreign keys, and other database elements). After that, we can use Oracle Apex wizards to create forms, reports, graphs and other web application pages. In our use case definition document, we have defined the following data entities and models:

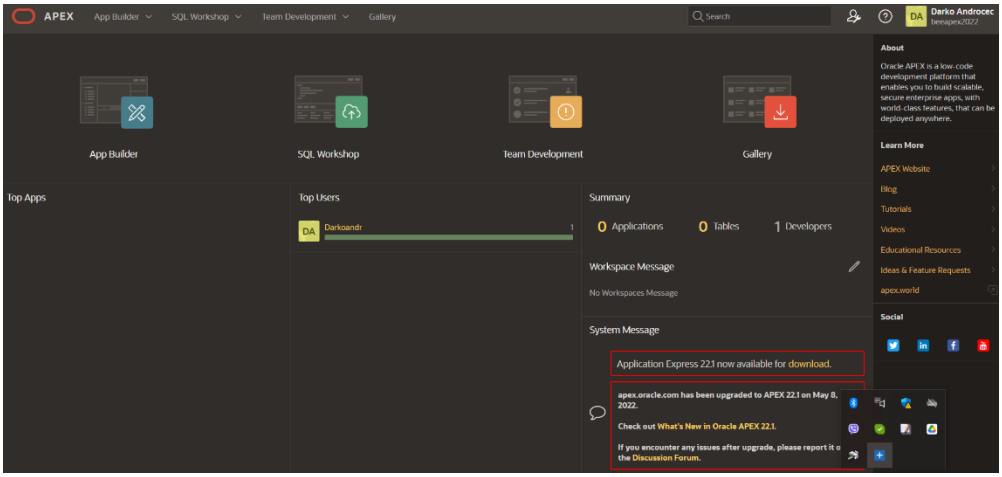
Logical Data Model



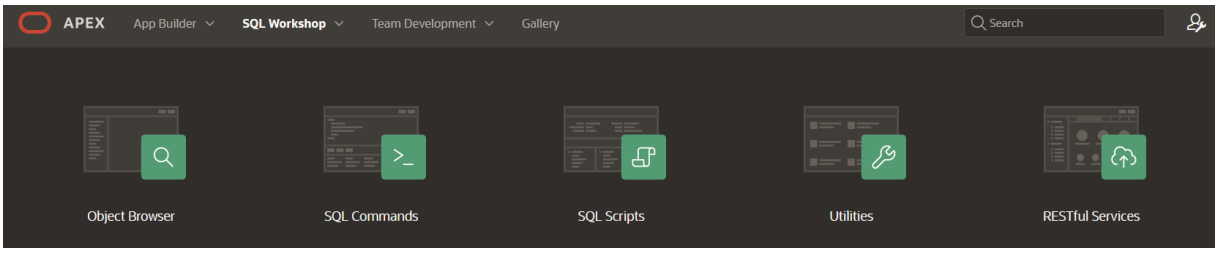
Relational Data Model



Next step is to create tables in Oracle Apex environment. Let's look again at the initial screen of the Oracle Apex:

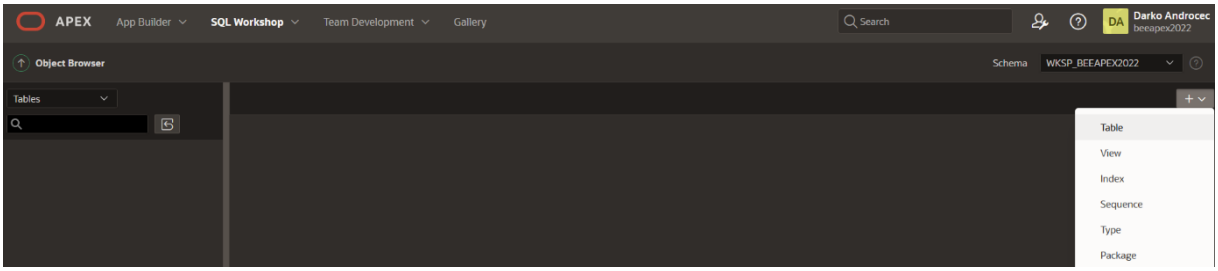


The part of the system related to database connection is SQL Workshop, so you click on it to navigate to the home screen for database creation:



The SQL Workshop is a collection of tools to manage database objects. SQL Workshop home page features the following icons: Object Browser; SQL Commands; SQL Scripts; Utilities; and RESTful Services. Object Browser enables developers to browse, create, and edit objects in a database in a graphical manner. It has many views defined to help you work with the underlying table structures. The views combine multiple base tables and use meaningful column names to minimize complexity and to be more intuitive to use. The SQL Commands tool enables you to enter ad-hoc SQL query. SQL Scripts enables you to store and run scripts. Apex provides utilities to build SQL queries, load and unload data from an Oracle database, generate DDL, view object reports, manage *User Interface Defaults*, restore dropped database objects, compare schemas, monitor the database, and view database details. RESTful Services enable the declarative specification of RESTful services used to access the database.

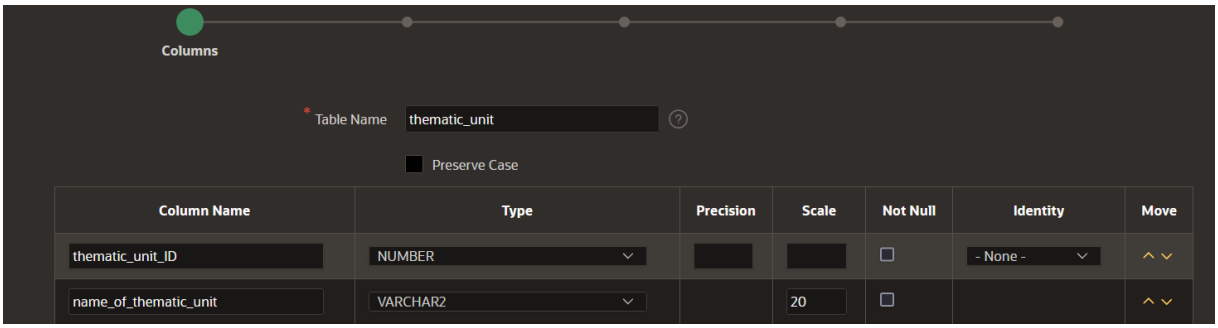
Here we will show how to use views of Object Browser to intuitively create tables for our use case (Browse catalog of plants). First, we will create a table for Thematic unit. Choose Object Browser clicking on + button in the upper right corner of the application and click on Table item:



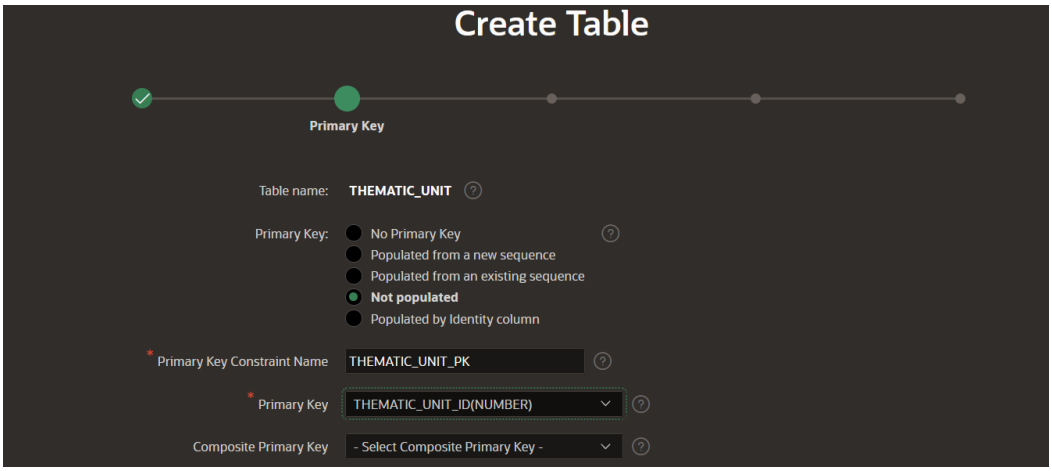
Create table: thematic_unit

On the first screen of the *Create Table* view you have to input table name, columns name and their types:

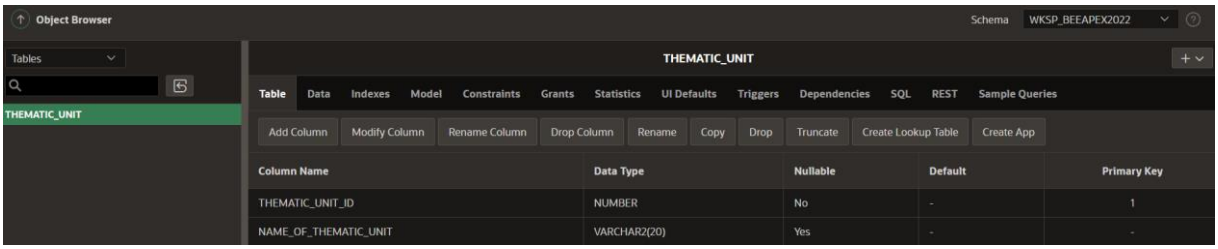
On the first screen of the *Create Table* view you have to input table name, columns name and their types:



Next screen deals with primary key, so we will select the following options:



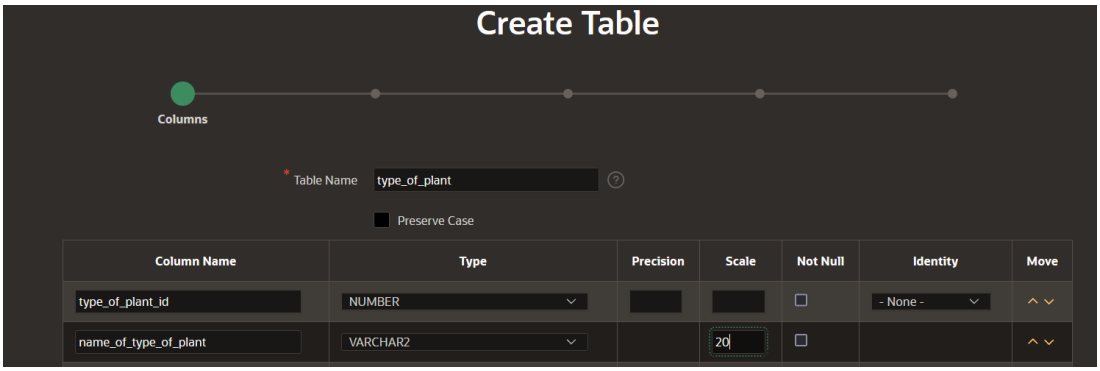
On the next two screen you can choose foreign keys, check condition on specific column or unique column definitions. Our table *thematic_unit* does not need any of the mentioned options, so we just skip these steps by clicking *Next* button, and choose the *Create Table* button to create this table:

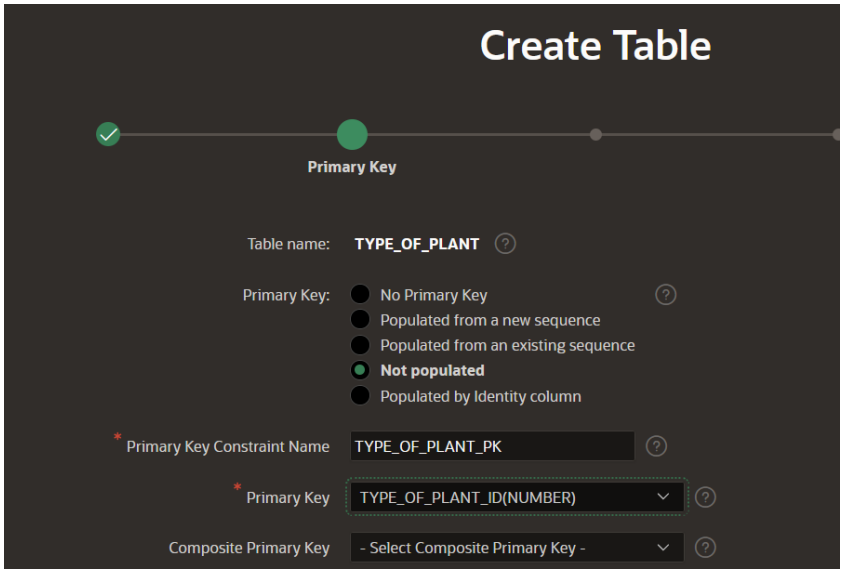


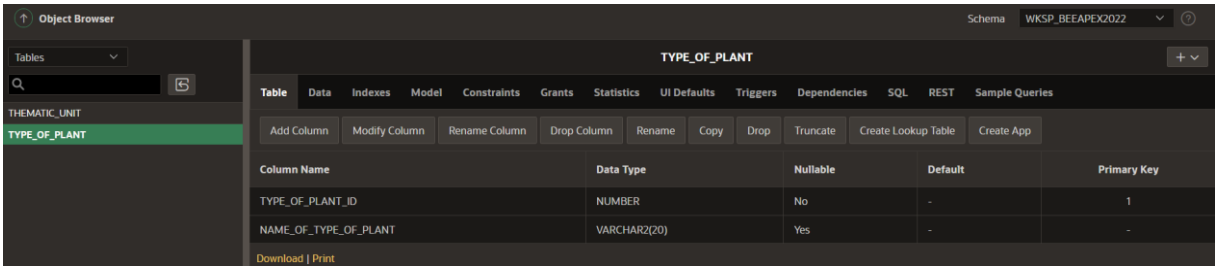
The table is created, and we are redirected to the screen where we can modify the table or fill in the values.

Similar steps are followed to create other tables from our ERA model:

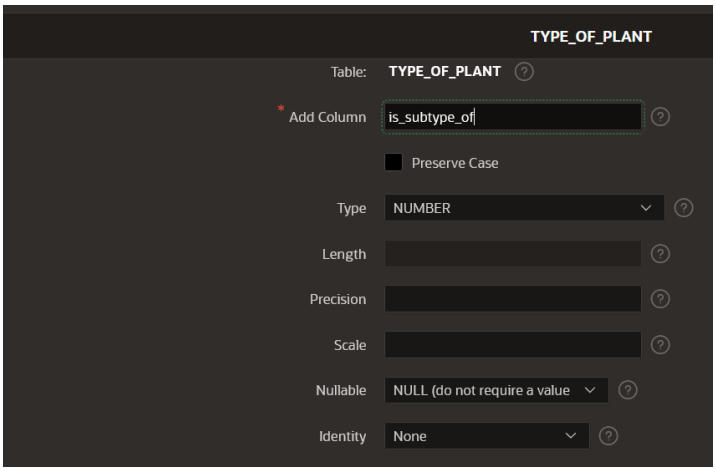
Create table type_of_plant

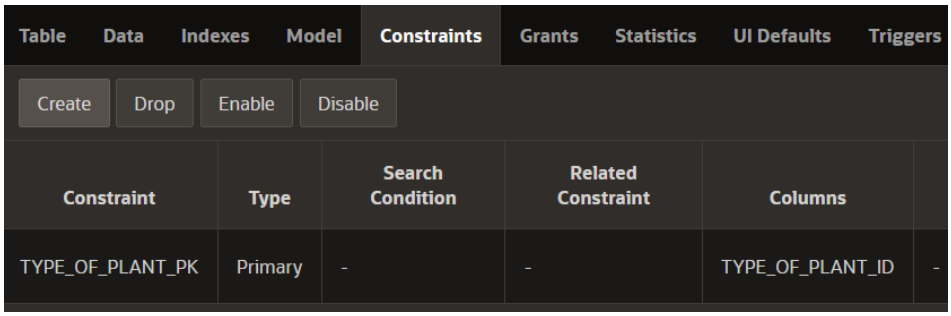


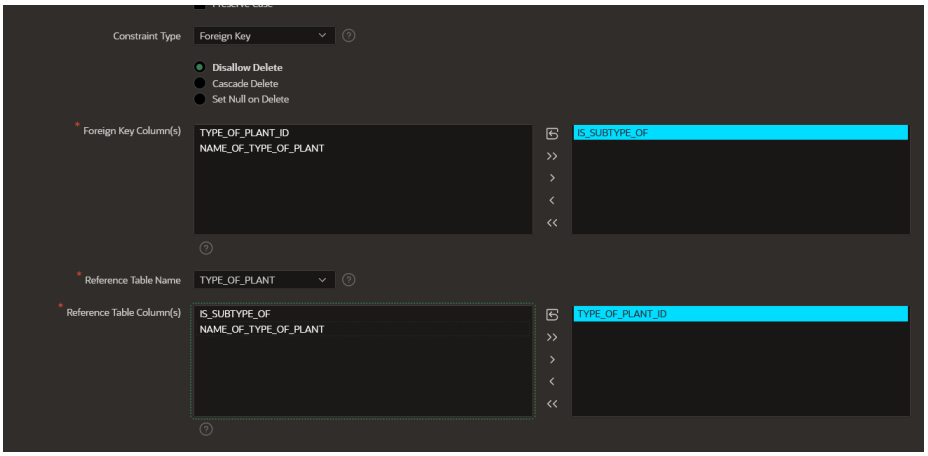




On ERA model of the Browse catalog use case you can see that some type can be subtype of different type of plant. We need to create this subsequently, because it is a foreign key to the same table and creation via *Create Table* view of *Object Browser* is not possible, because its graphical interface expect that you have already created table on which your foreign key connects. So here we use Add Column and Create Constraints button to enable plant subtypes:

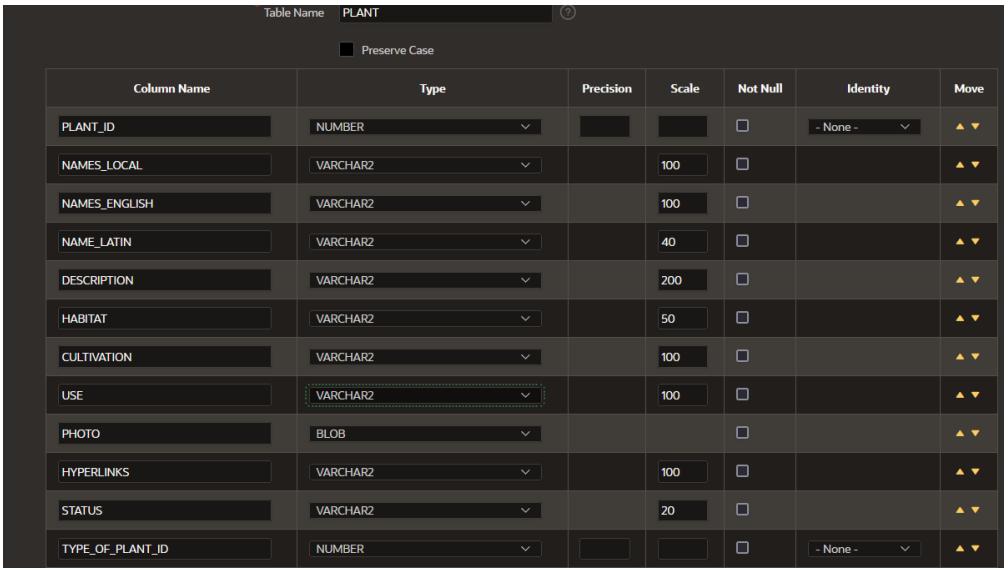


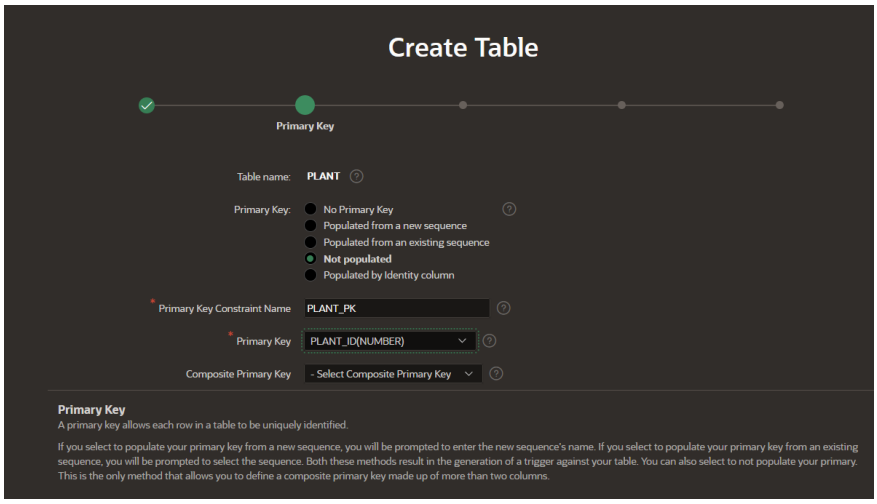




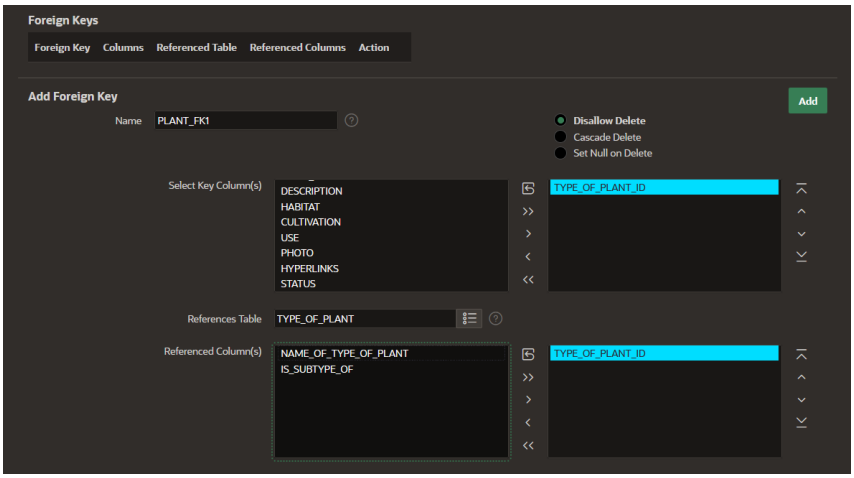
Create table plant

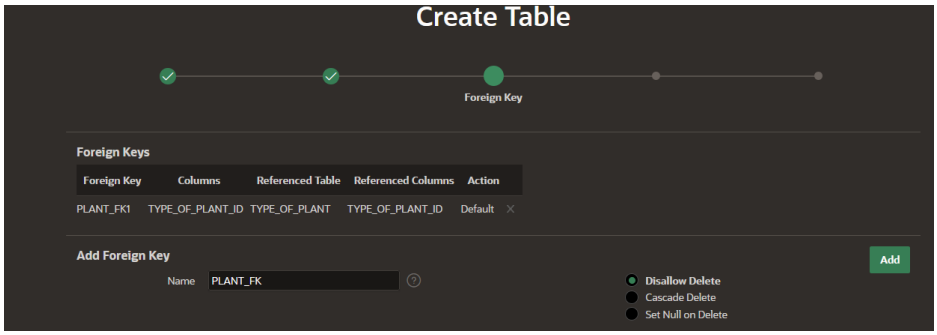
The table plant is a central table of Browse catalog use case connected directly via foreign keys with other tables from our data model:





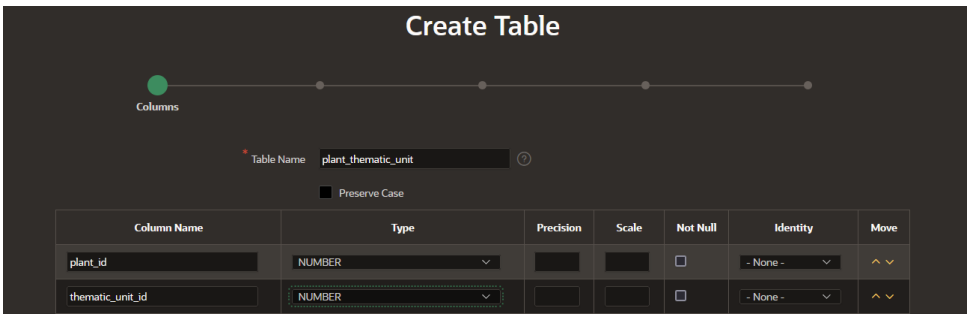
Here we have a foreign key on *type_of_plant* table. Do not forget to click on the Add button to create the foreign key (It should be visible in the Foreign Keys list on the end):



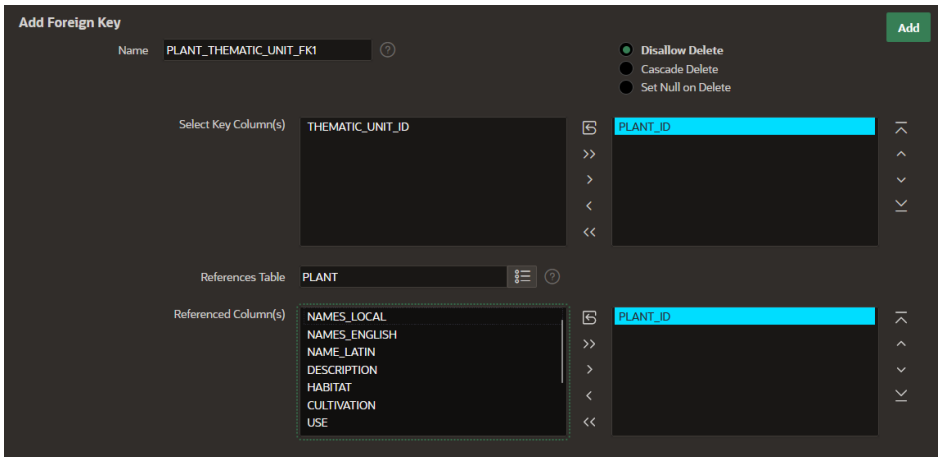


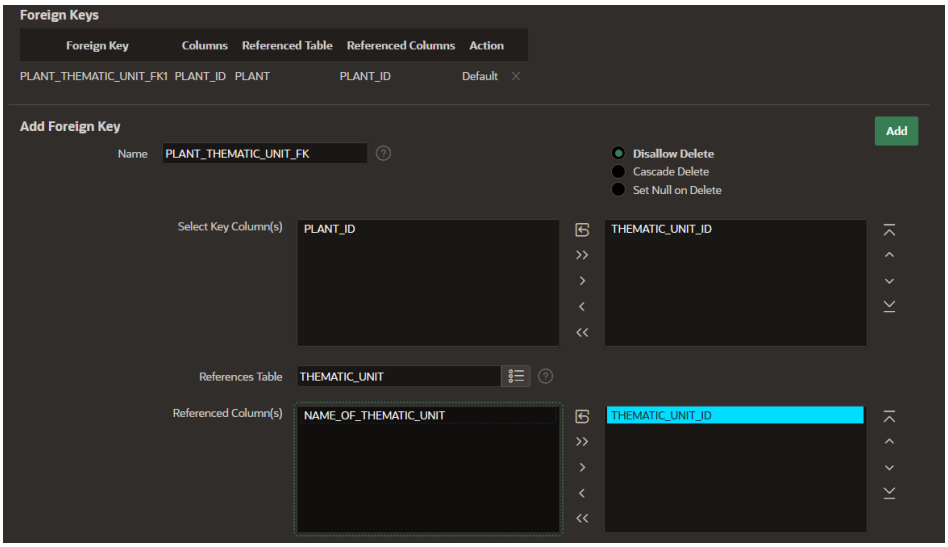
Create table plant_thematic_unit

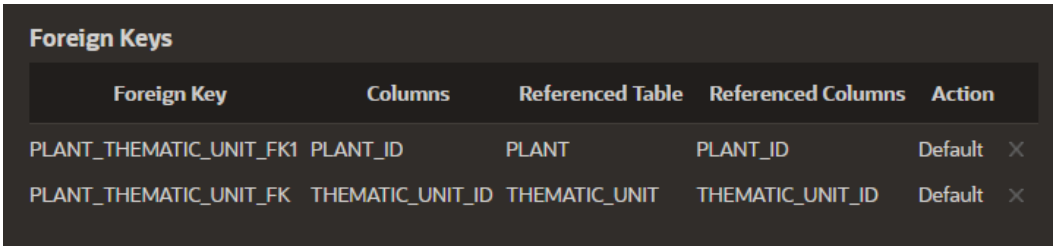
The connection between the table *plant* and the table *thematic_unit* is many-to-many, so we will create the associate table called *plant_thematic_unit* between them to implement this relation (M:M).



In this associated table, primary key has two components (*plant_id* and *thematic_unit_id*), with each of components referencing to a corresponding table (two foreign keys).



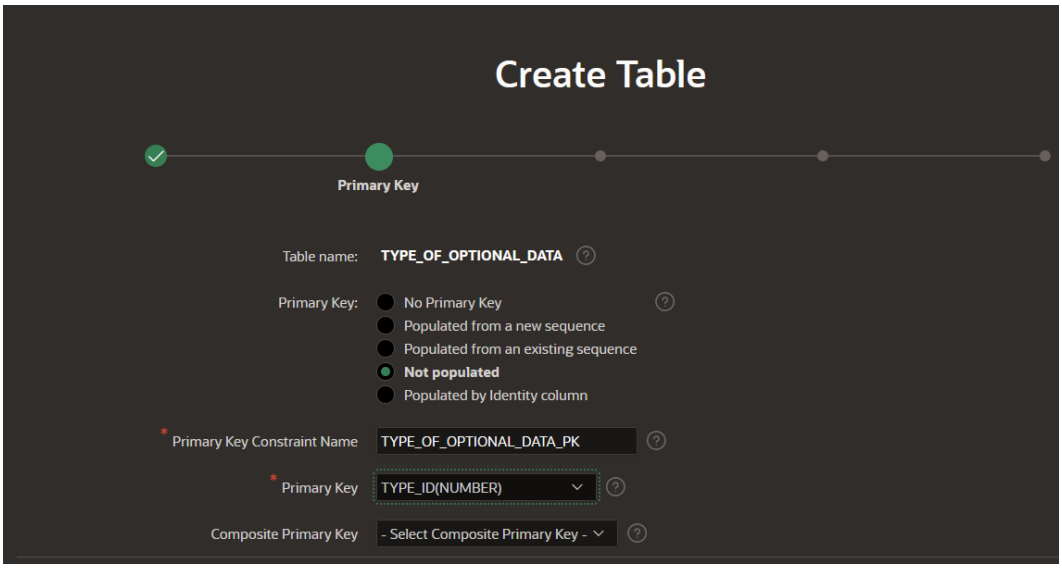




Using the same wizards, the optional tables (plant_optional_data, type_of_optional_data, user) are also created:

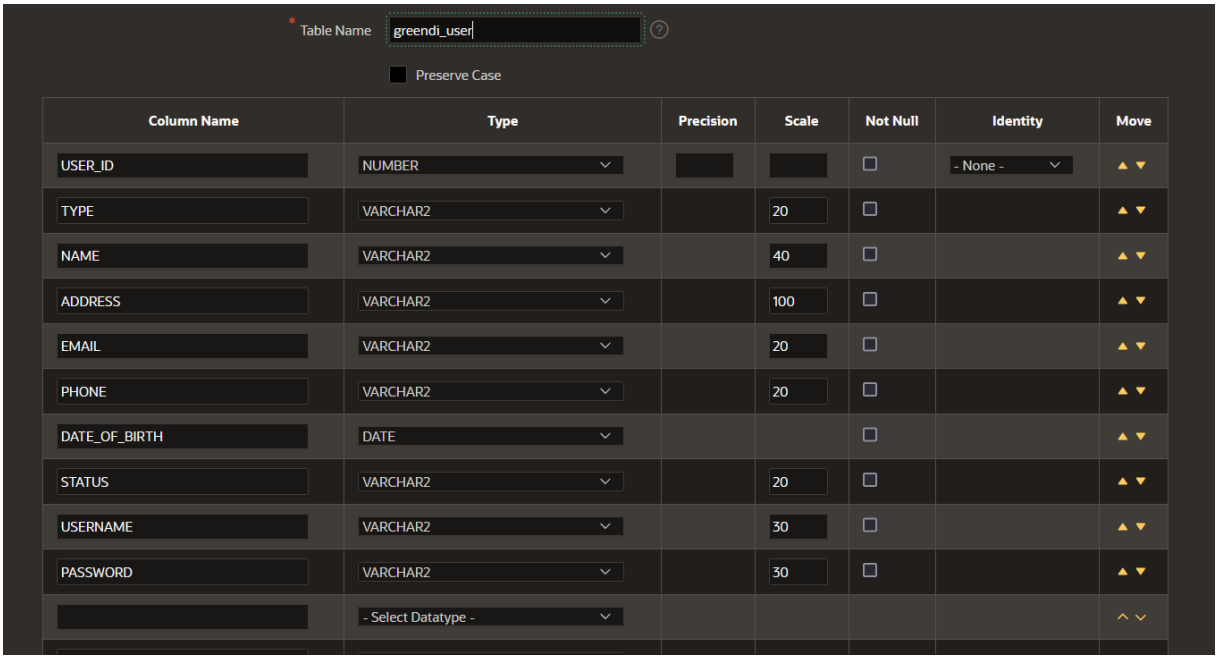
Create table type_of_optional_data

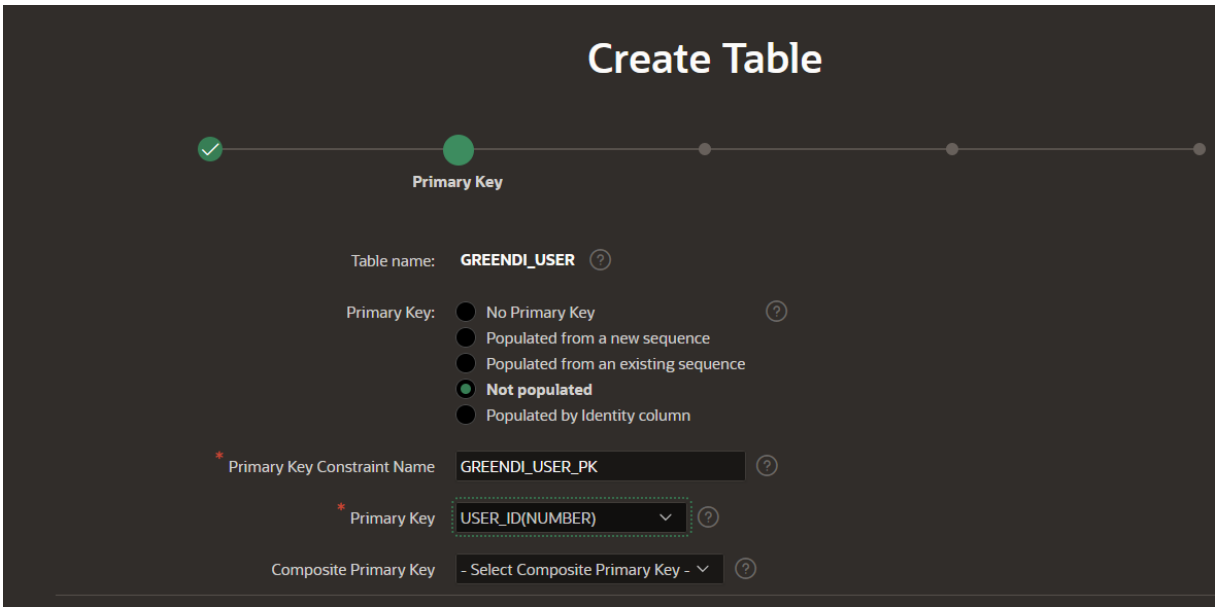




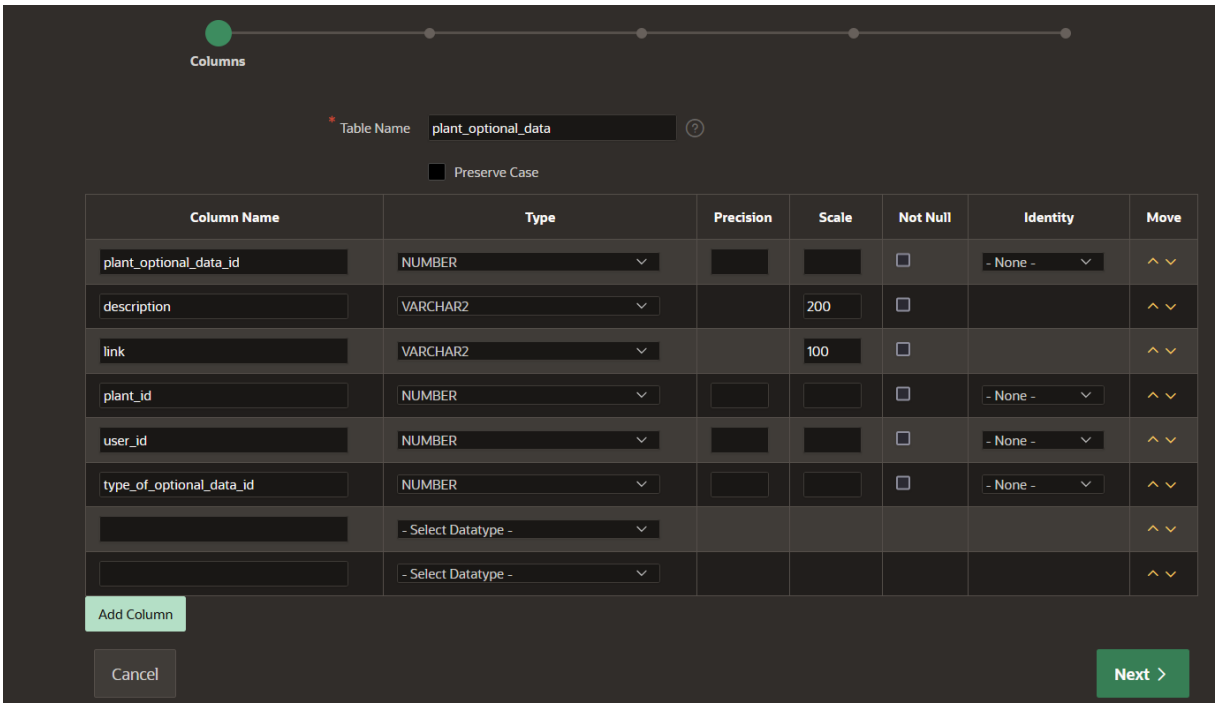
Create table greendi_user

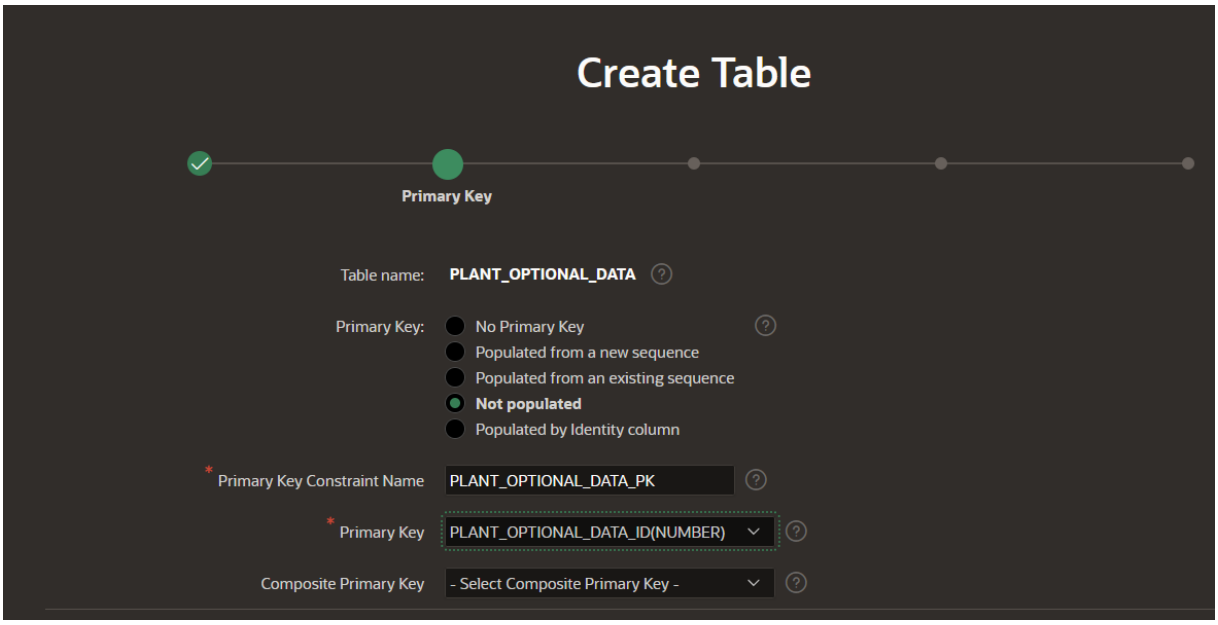
Note: *User* is reserved table name in Oracle Apex, so we use here *greendi_user* as table name.



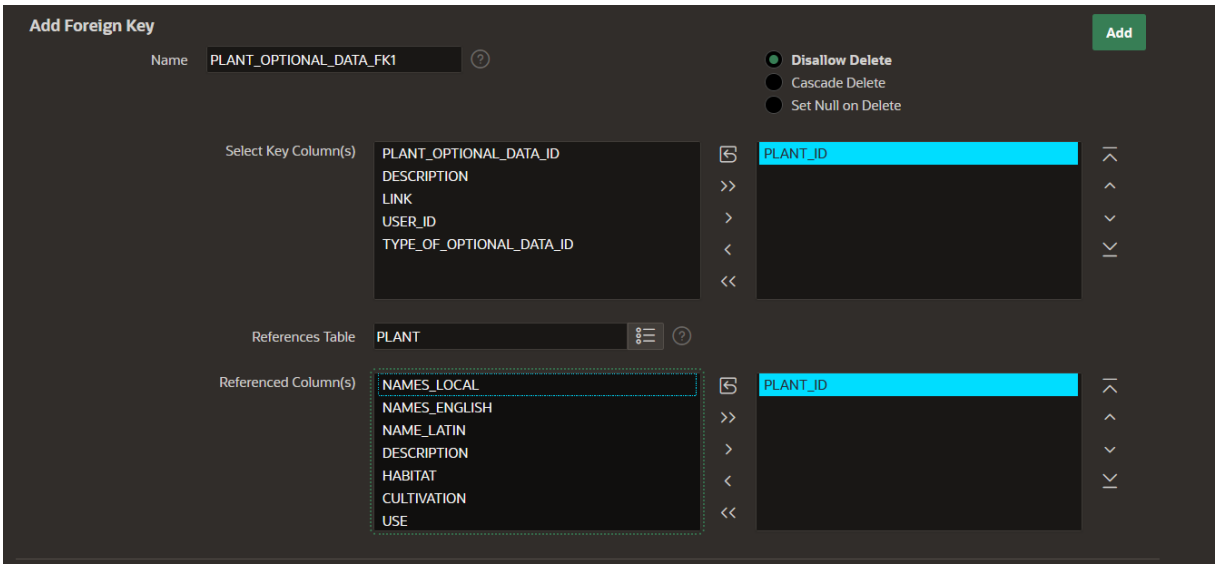


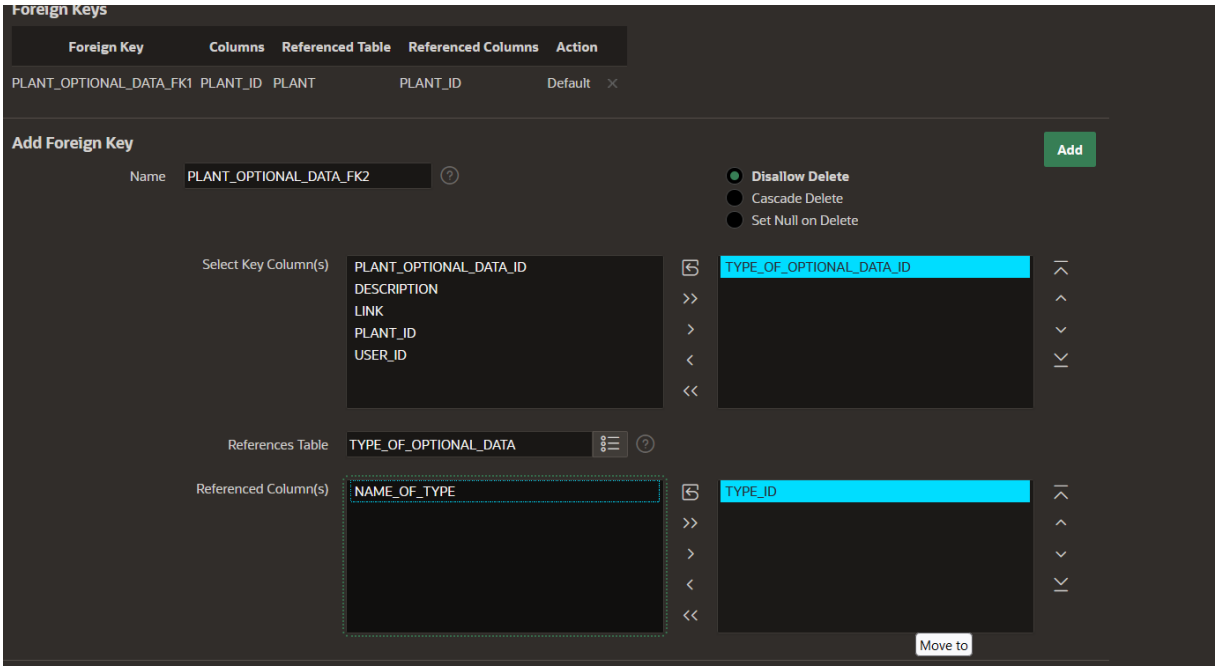
Create table plant_optional_data

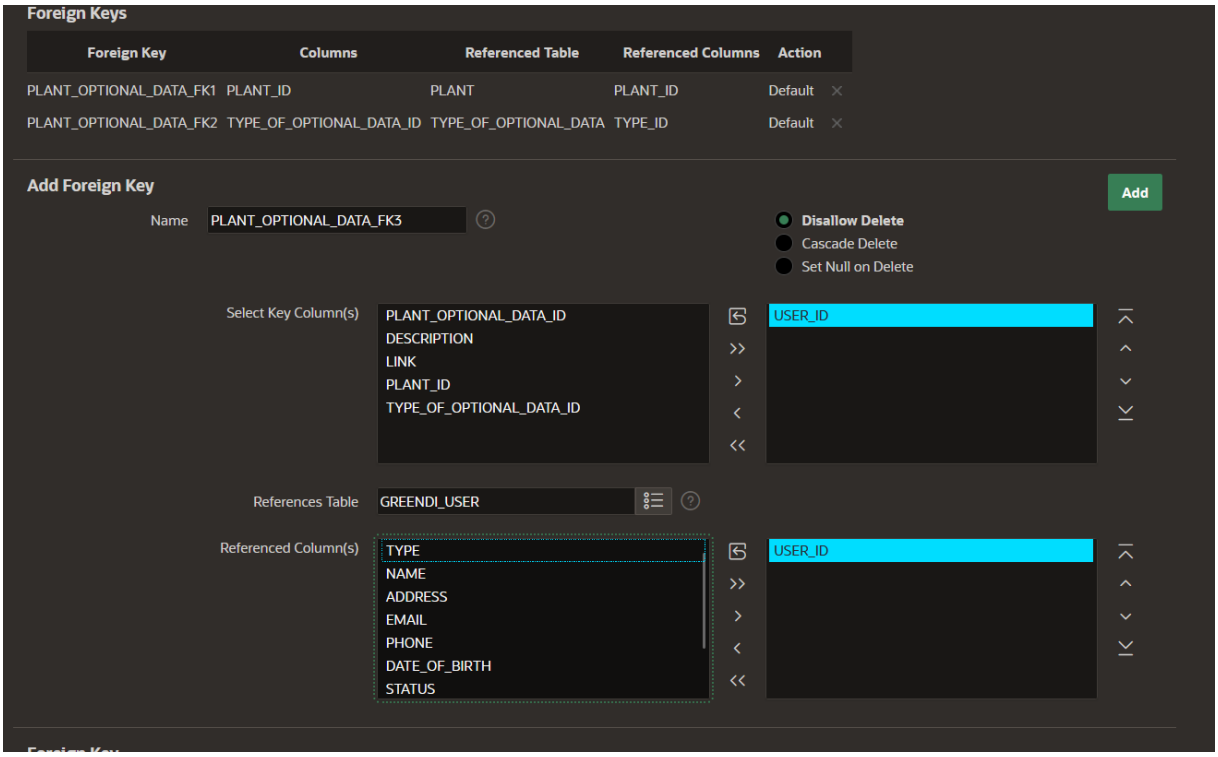




Three defined foreign keys:

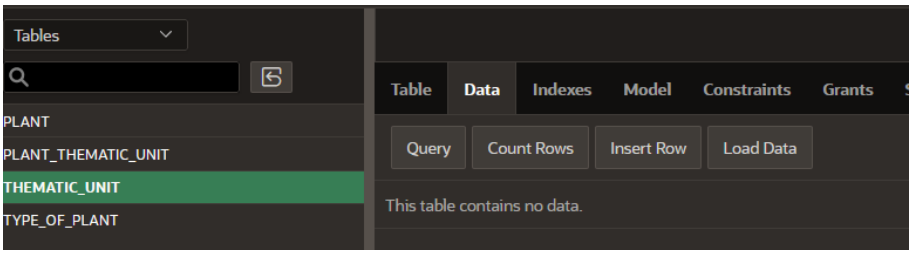


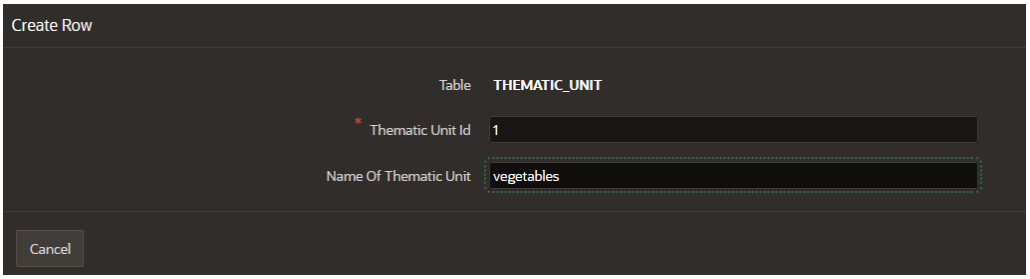


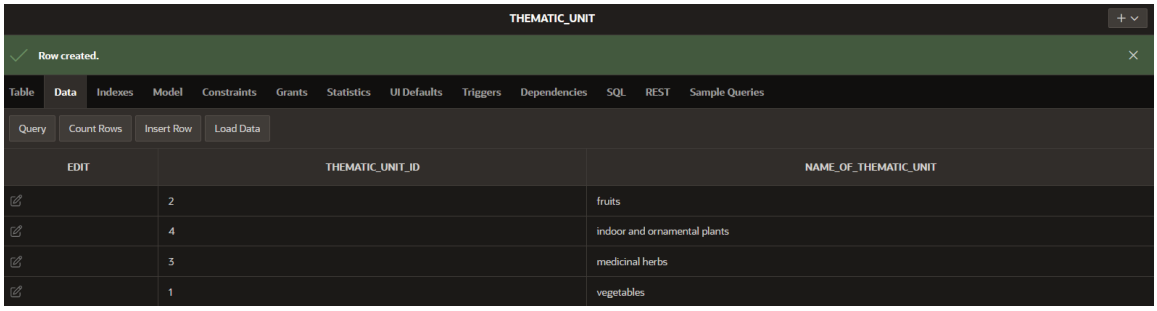


Filling in tables with data

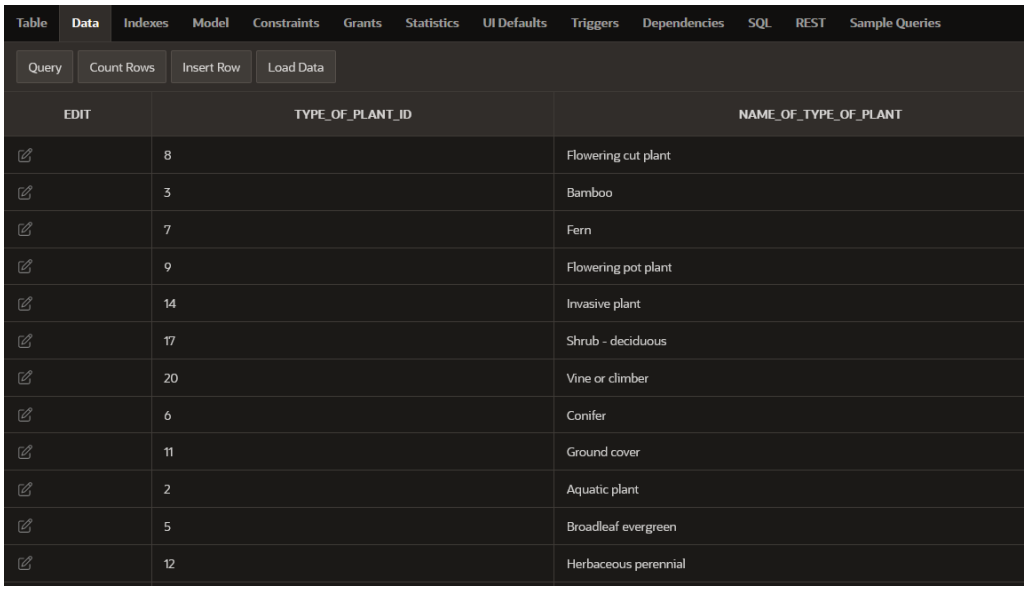
In the next steps we will fill each table with sample values. This is good practice, because we can notice quicker eventual errors after initial web application generation. In *Object Browser*, we click on the table that we want to fill with data and click on *Data* tab. Next, we click on *Insert Row* button and add some data into each table from our data model.







You can now use some publicly available plant databases to fill-in some values. For example, using data from https://plantdatabase.kpu.ca/ you can fill some values in the main tables from our model:



PLANT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

Query | Count Rows | Insert Row | Load Data

EDIT	PLANT_ID	NAMES_LOCAL	NAMES_ENGLISH	NAME_LATIN	DESCRIPTION	HABITAT	CULTIVATION	USE	PHOTO	HYPERLINKS	STATUS	TYPE_OF_PLANT_ID
	1	hardy kiwi	hardy kiwi	Actinidia arguta	Leaves usually ovate, sometimes ovate-oblong, 6-12 x 5-10 cm, margin sharply serrate, apex abruptly acuminate; florets greenish yellow or white	disturbed habitats and forest edges	Softwood cuttings, Grafting	-	[datatype]	https://plantdatabase.kpu.ca/plant/plantDetail/1482	-	20
	2	raspberry	raspberry	Rubus idaeus	Winter: multiple straight basal stems with small prickles, Summer:same with compound leaves	forest clearings and edges, disturbed areas	Cuttings, Hardwood cuttings, Layering	-	[datatype]	https://plantdatabase.kpu.ca/plant/plantDetail/1369	-	20

PLANT_THEMATIC_UNIT

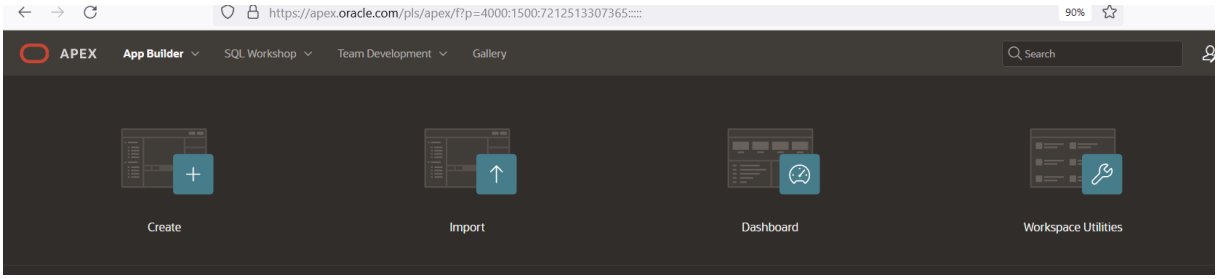
Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

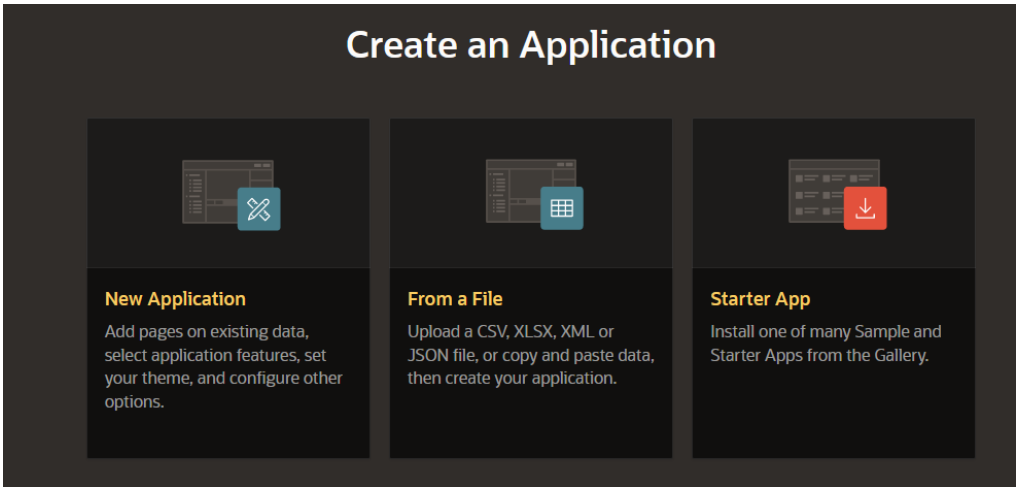
Query | Count Rows | Insert Row | Load Data

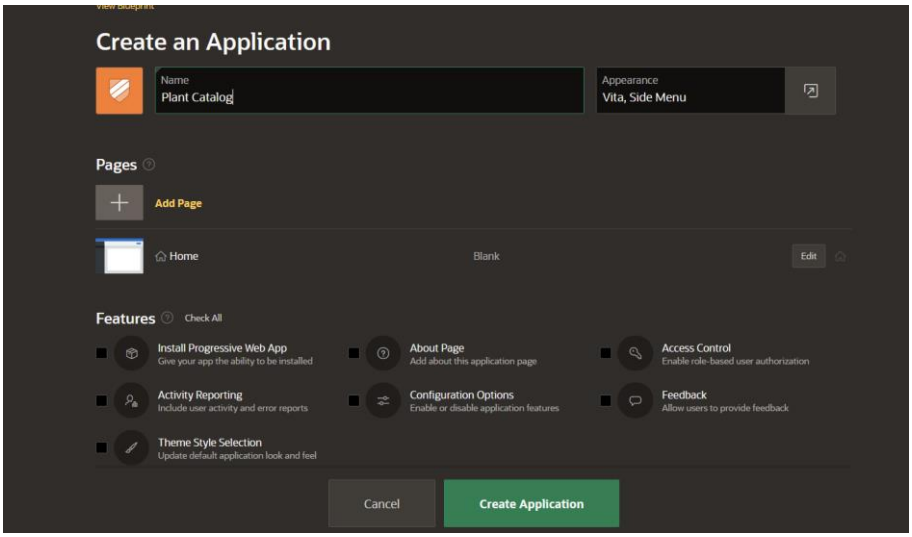
EDIT	PLANT_ID	THEMATIC_UNIT_ID
	1	2
	2	2

Creating an Oracle Apex application – public part – open browsing of plant data

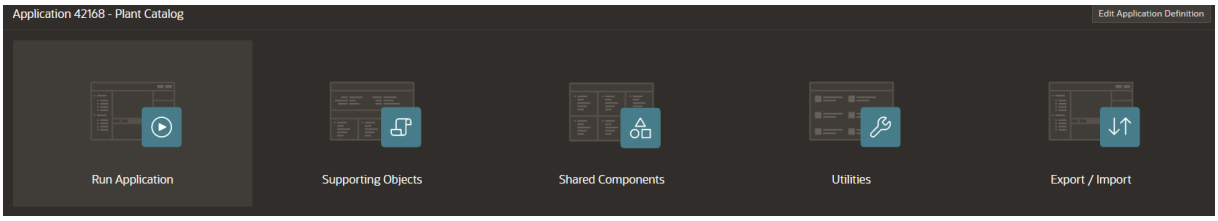
Next step is to create/generate an Oracle Apex application on the top of the defined database. On the Oracle Apex home screen you choose A*pp Builder* and then *Create*. For the application name, you can input *Plant Catalog* and leave all other options default.



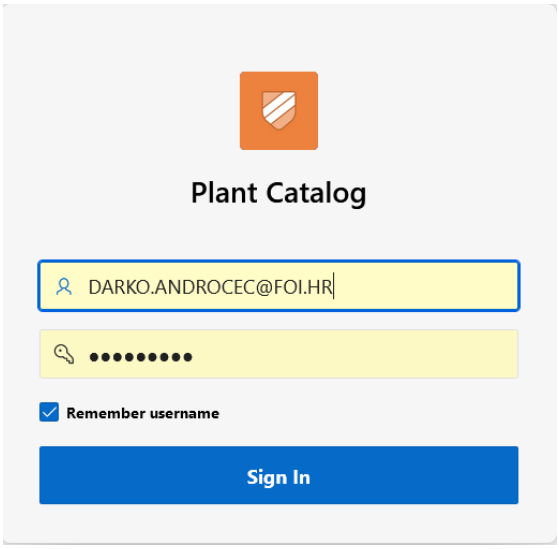




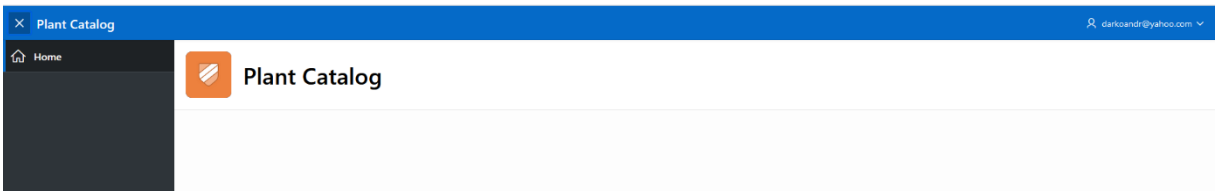
Web application with home page is now created, and you can click on Run Application button to execute it and see in the web browser environment:



By default, and Oracle Apex account authentication is selected, so you need to fill in you Oracle Apex username and password to access the new application:

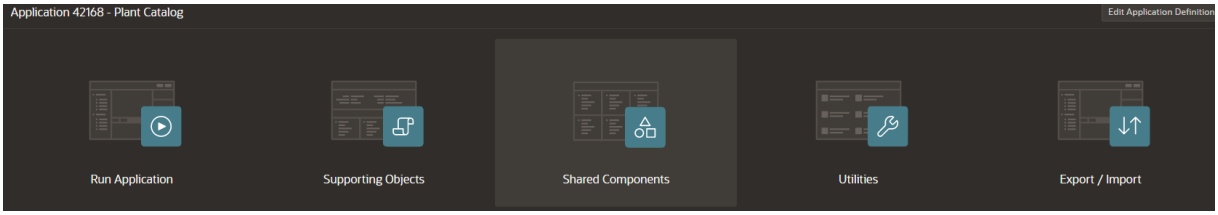


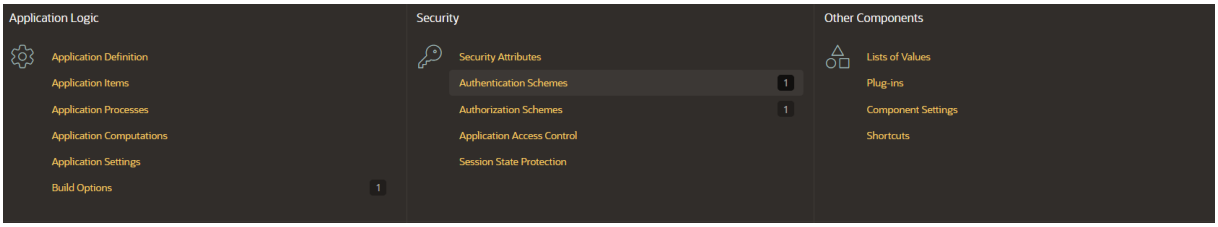
Next, the home page of the application is opened. It is for now empty, and our application does not have any further pages for now:

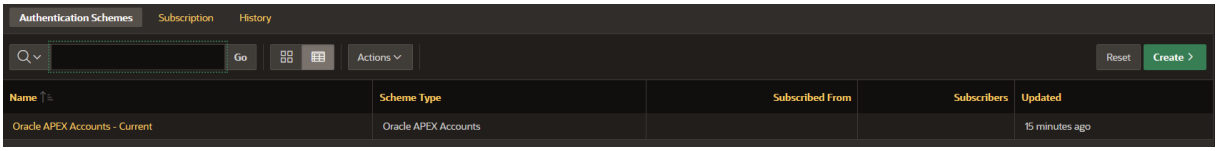


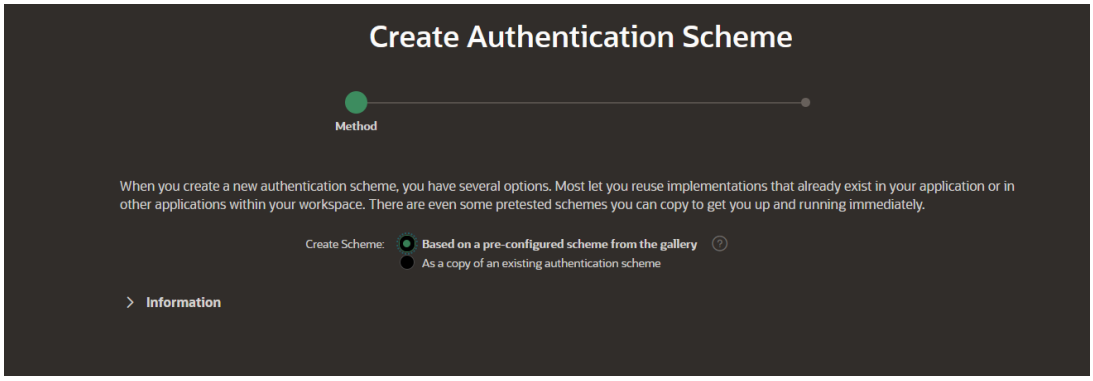
Our plant catalog should be publicly available so we will remove authentication for this application, so everyone can access it. In the application administration part of Oracle Apex we choose *Shared components and Authentication Schemes* in the *Security* widget. Then we create new current authentication setup where we enable everyone with application link to access this application.

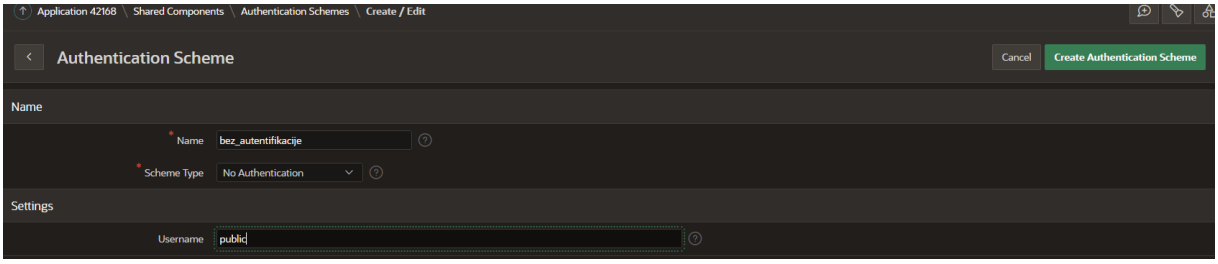










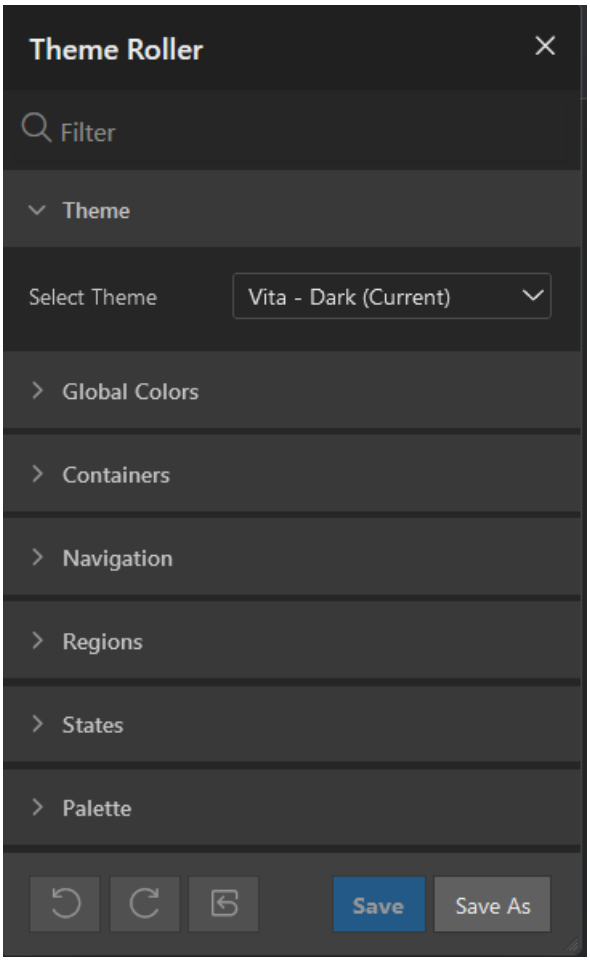


Now the application can be accessed without user registration.

You can now run application and look at it. Bellow there is a widget with which you can change visually the look of the home screen of application using the following options: *QuickEdit*, *Customize* (*Theme Roller*, *Edit Logo*, *Edit App Icon* etc.). Try to use these options to change the application theme and user interface of home screen.

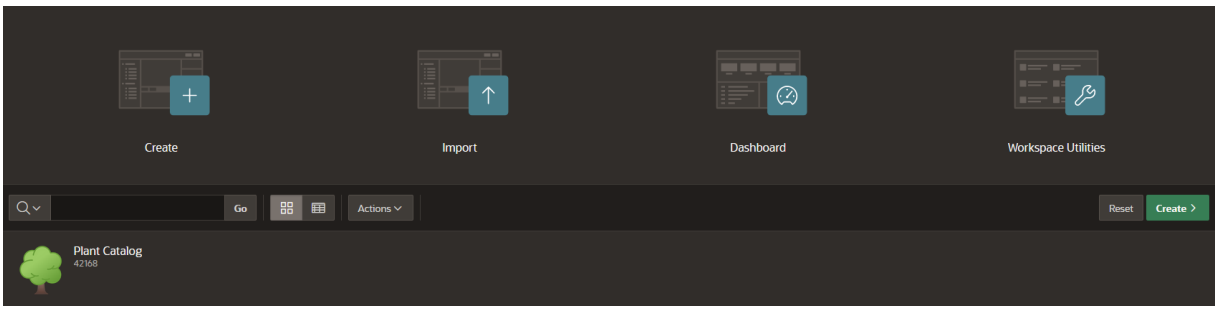


For example, we will explain how to change application theme and how to add background picture to the home screen. The theme can be changed in the *Customize – Theme Roller – Select Theme* (here we have chosen *Vita-Dark* theme, but you can also try other available themes):

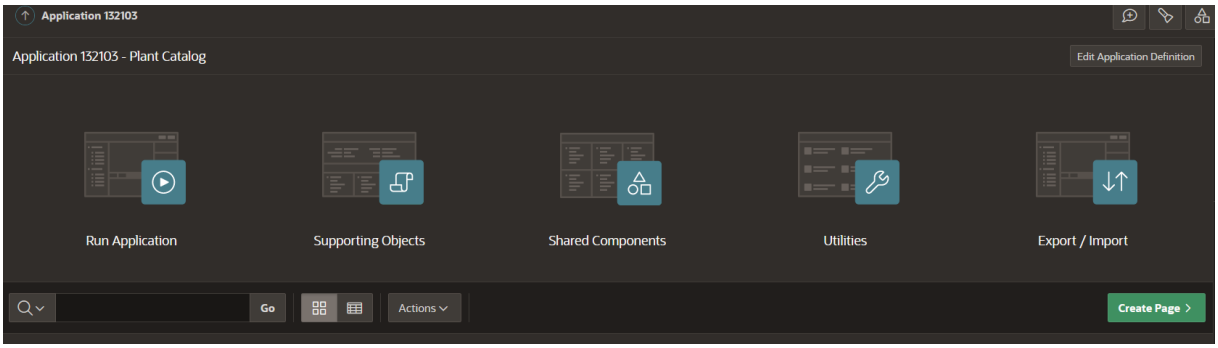


Creating simple interactive reports

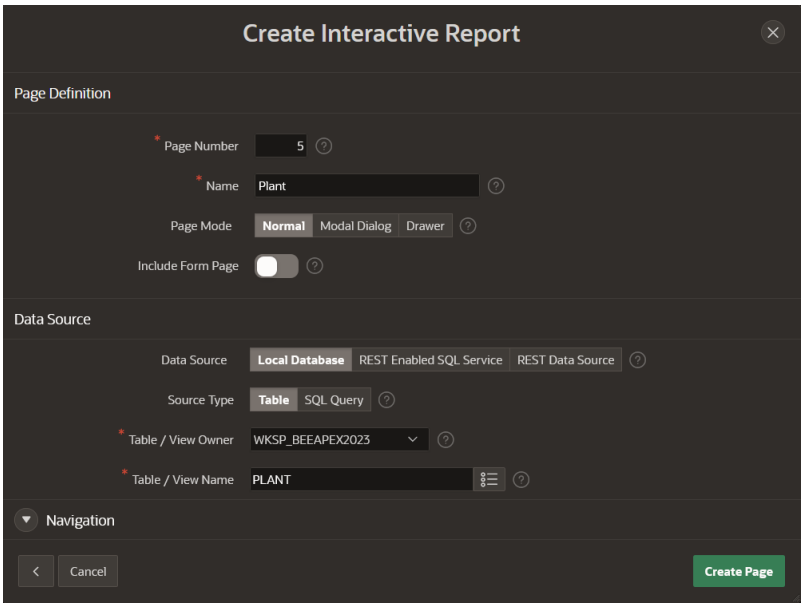
Next we can create simple interactive reports for all tables from the database, so every visitors can take look at this data. We go to *App Builder* and choose our application (*Plant Catalog*):

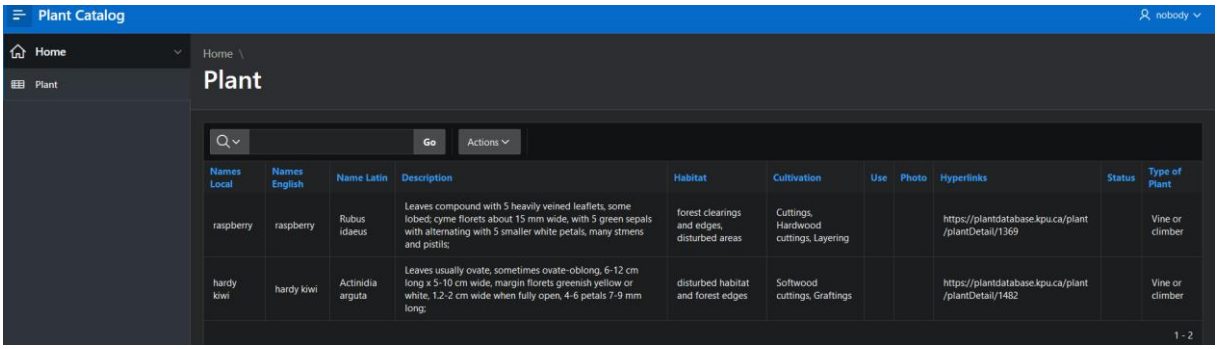


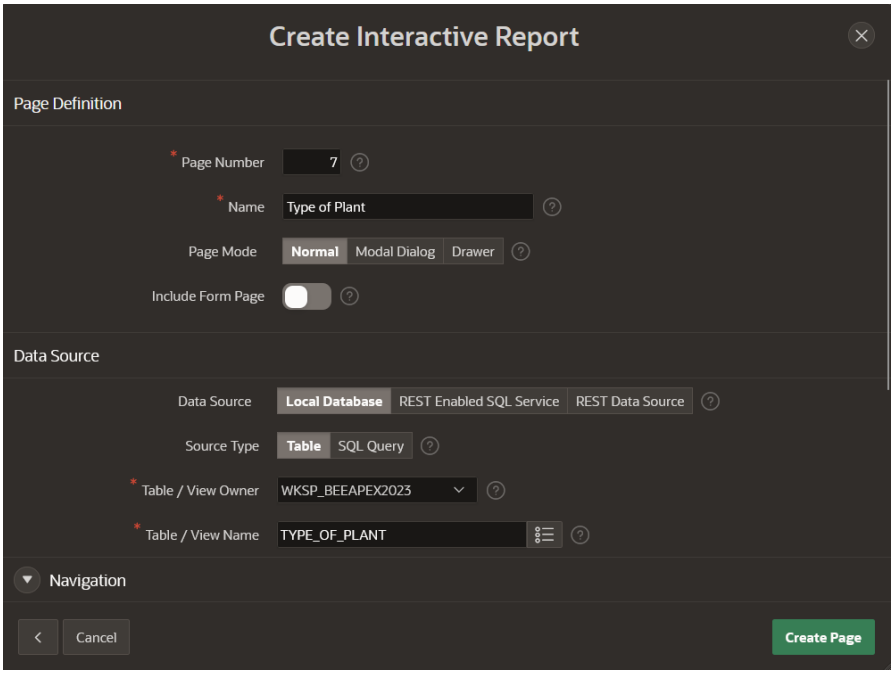
Next we use Create Page wizard to create the interactive reports for all tables from our database. The screens to do this are captured below.

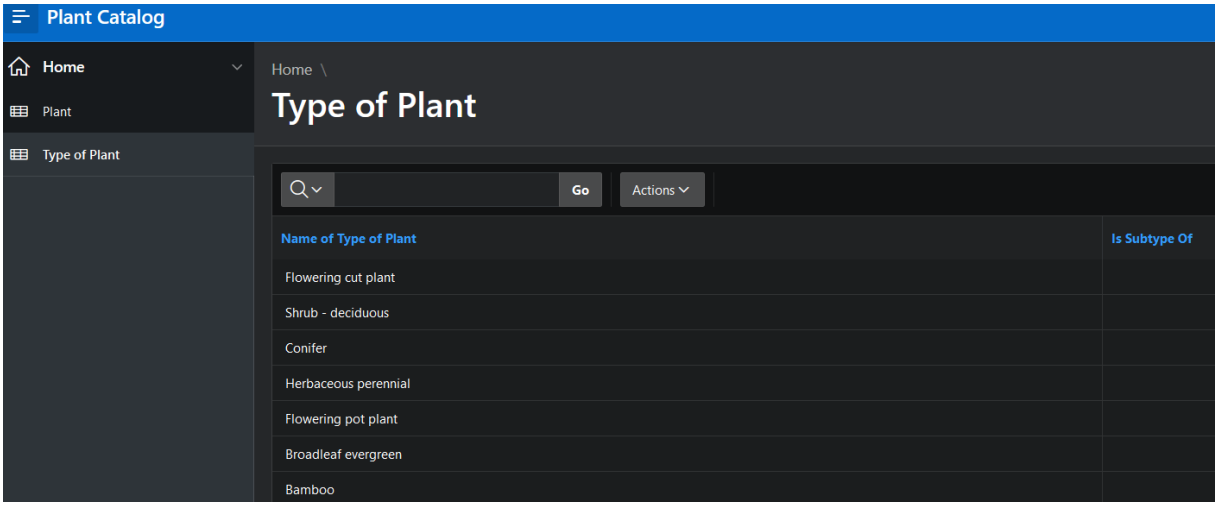


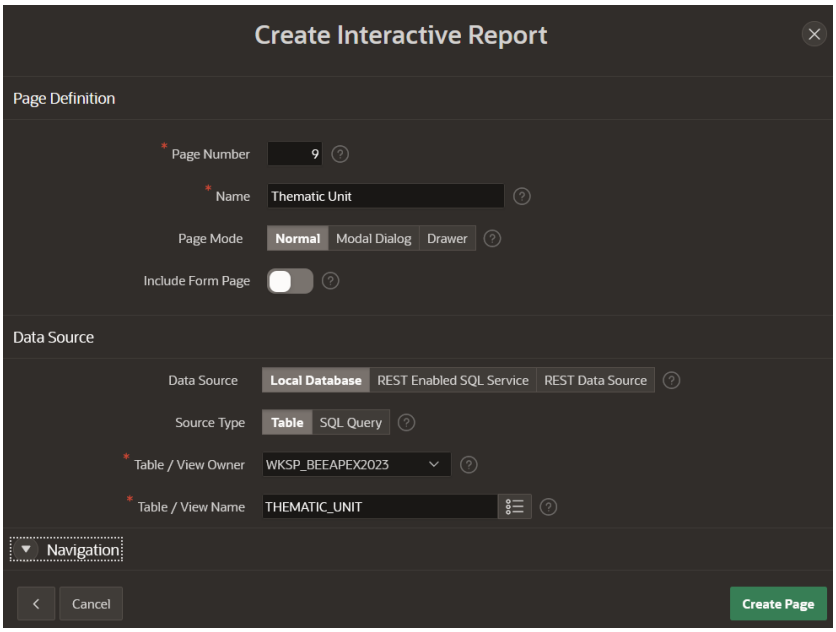


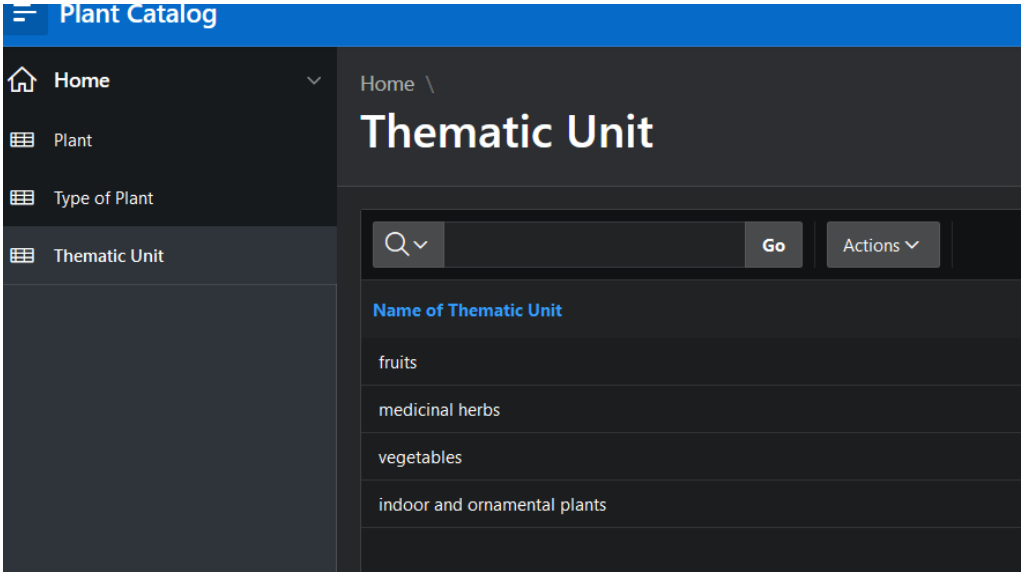


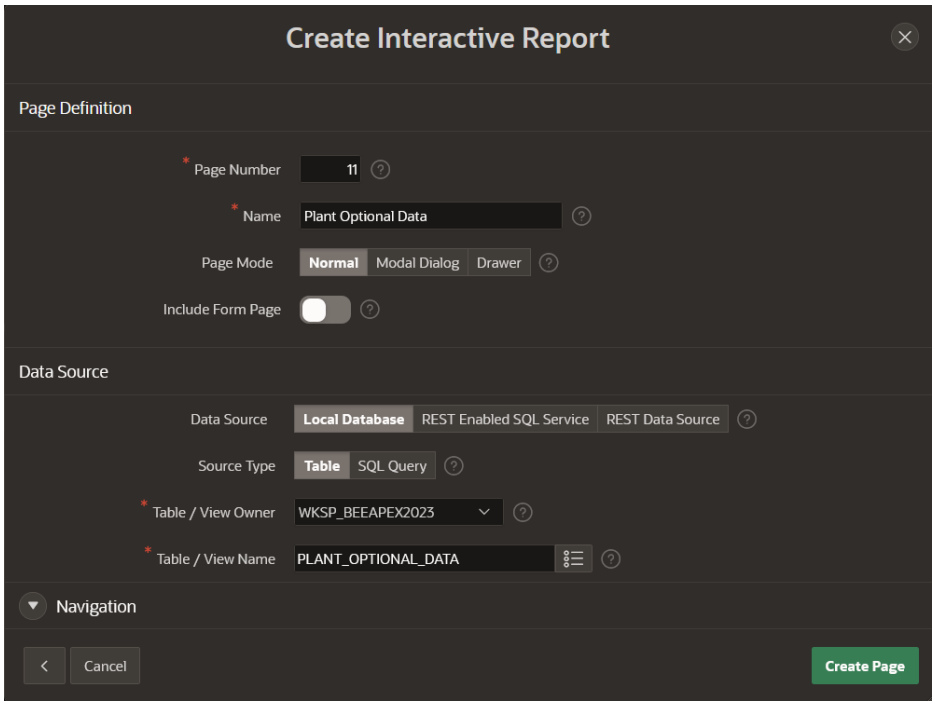

Plant Catalog
nobody
Home
Plant
Home \
Plant
Go
Actions
Names Local
Names English
Name Latin
Description
Habitat
Cultivation
Use
Photo
Hyperlinks
Status
Type of Plant
raspberry
raspberry
Rubus idaeus
Leaves compound with 5 heavily veined leaflets, some lobed; cyme florets about 15 mm wide, with 5 green sepals with alternating with 5 smaller white petals, many stmens and pistils;
forest clearings and edges, disturbed areas
Cuttings, Hardwood cuttings, Layering
https://plantdatabase.kpu.ca/plant/plantDetail/1369
Vine or climber
hardy kiwi
hardy kiwi
Actinidia arguta
Leaves usually ovate, sometimes ovate-oblong, 6-12 cm long x 5-10 cm wide, margin florets greenish yellow or white, 1.2-2 cm wide when fully open, 4-6 petals 7-9 mm long;
disturbed habitat and forest edges
Softwood cuttings, Graftings
https://plantdatabase.kpu.ca/plant/plantDetail/1482
Vine or climber
1 - 2

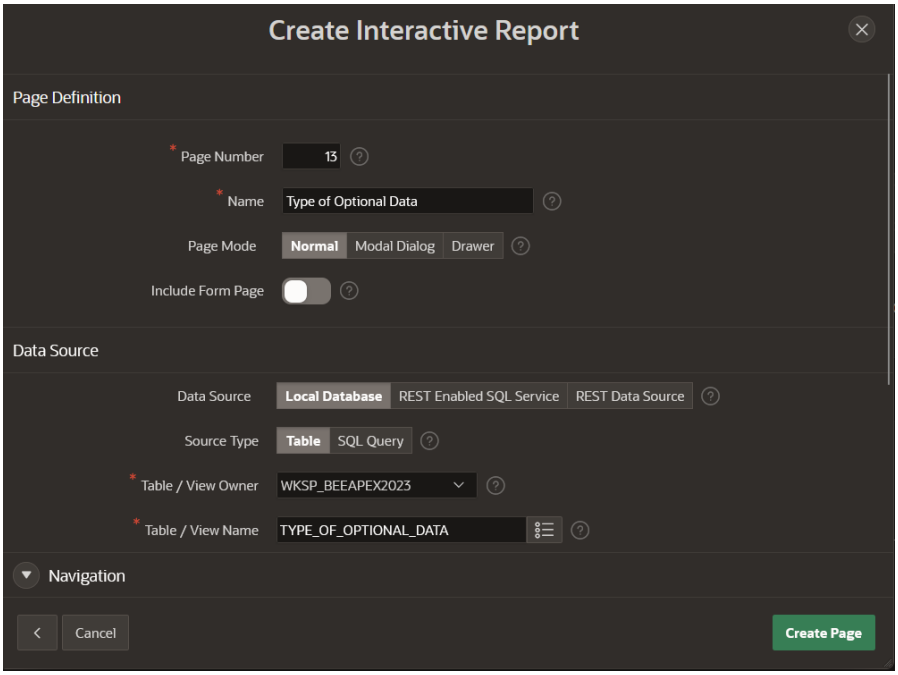

Create Interactive Report
Page Definition
* Page Number
7
* Name
Type of Plant
Page Mode
Normal
Modal Dialog
Drawer
Include Form Page
Data Source
Data Source
Local Database
REST Enabled SQL Service
REST Data Source
Source Type
Table
SQL Query
* Table / View Owner
WKSP_BEEAPEX2023
* Table / View Name
TYPE_OF_PLANT
Navigation
Cancel
Create Page

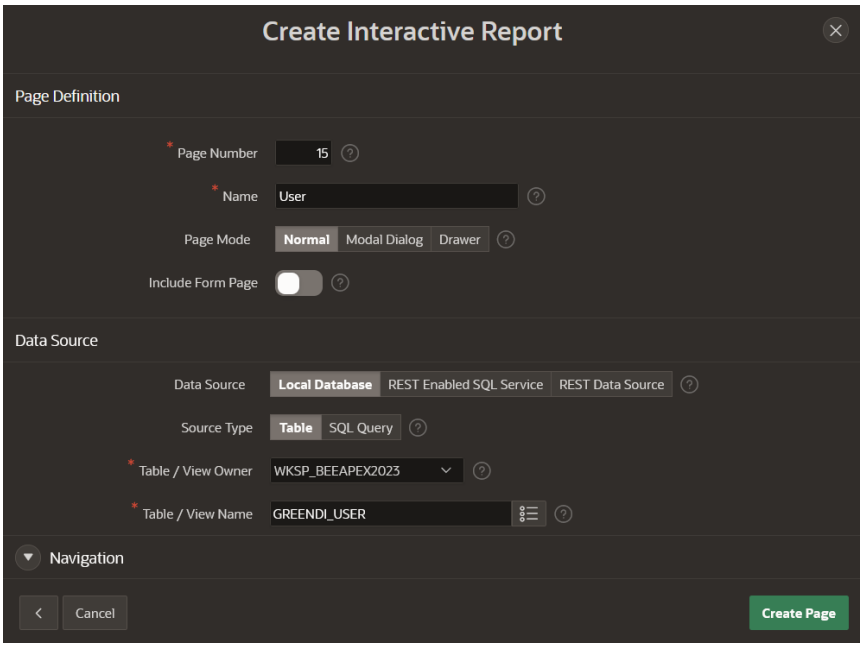

Plant Catalog
Home
Plant
Type of Plant
Home \
Type of Plant
Go
Actions
Name of Type of Plant
Is Subtype Of
Flowering cut plant
Shrub - deciduous
Conifer
Herbaceous perennial
Flowering pot plant
Broadleaf evergreen
Bamboo


Create Interactive Report
Page Definition
Page Number
9
Name
Thematic Unit
Page Mode
Normal
Modal Dialog
Drawer
Include Form Page
Data Source
Data Source
Local Database
REST Enabled SQL Service
REST Data Source
Source Type
Table
SQL Query
Table / View Owner
WKSP_BEEAPEX2023
Table / View Name
THEMATIC_UNIT
Navigation
Cancel
Create Page


Plant Catalog
Home
Plant
Type of Plant
Thematic Unit
Home \
Thematic Unit
Go
Actions
Name of Thematic Unit
fruits
medicinal herbs
vegetables
indoor and ornamental plants

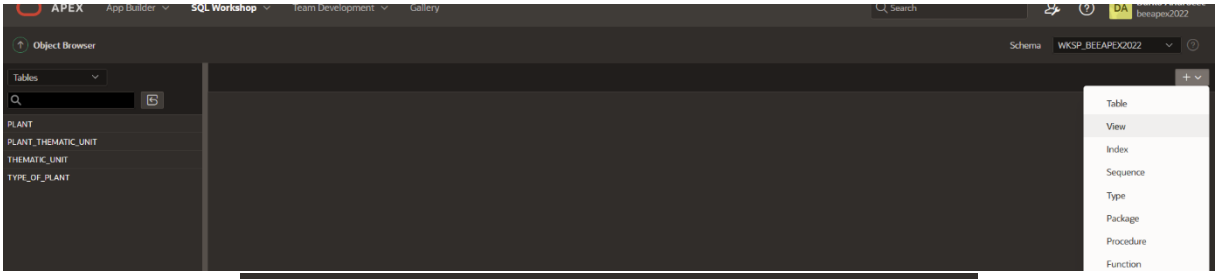

Create Interactive Report
Page Definition
* Page Number
11
* Name
Plant Optional Data
Page Mode
Normal
Modal Dialog
Drawer
Include Form Page
Data Source
Data Source
Local Database
REST Enabled SQL Service
REST Data Source
Source Type
Table
SQL Query
* Table / View Owner
WKSP_BEEAPEX2023
* Table / View Name
PLANT_OPTIONAL_DATA
Navigation
Cancel
Create Page

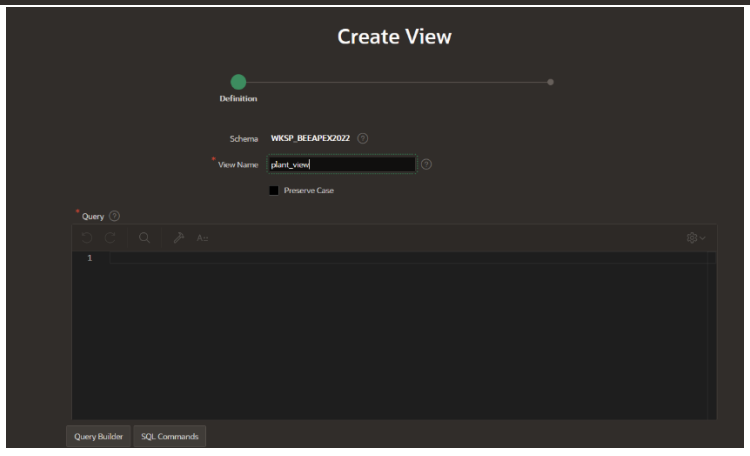

Create Interactive Report
Page Definition
* Page Number
13
* Name
Type of Optional Data
Page Mode
Normal
Modal Dialog
Drawer
Include Form Page
Data Source
Data Source
Local Database
REST Enabled SQL Service
REST Data Source
Source Type
Table
SQL Query
* Table / View Owner
WKSP_BEEAPEX2023
* Table / View Name
TYPE_OF_OPTIONAL_DATA
Navigation
Cancel
Create Page

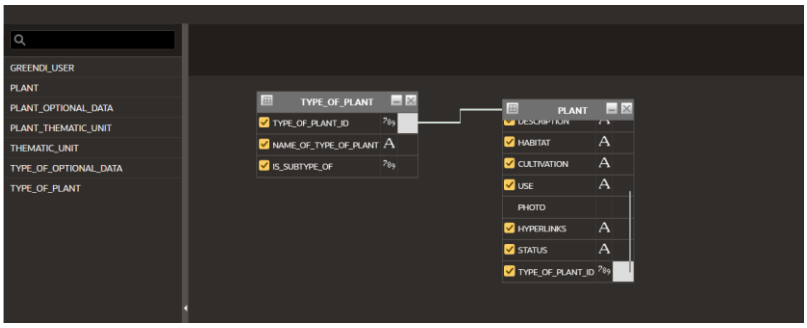


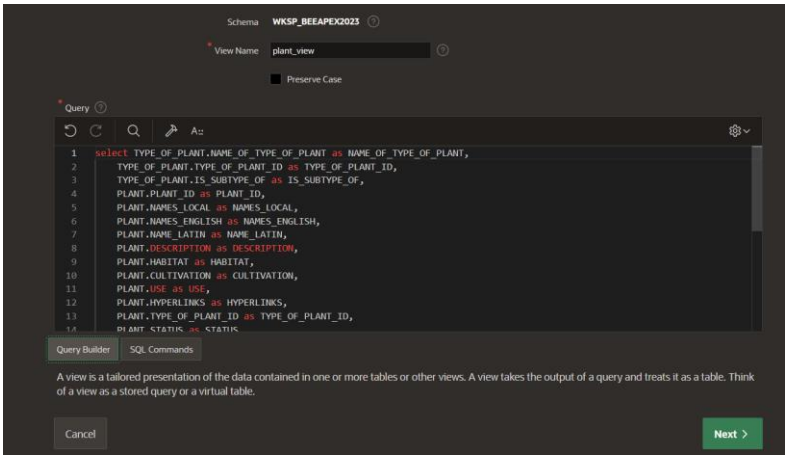
Creating views on plants

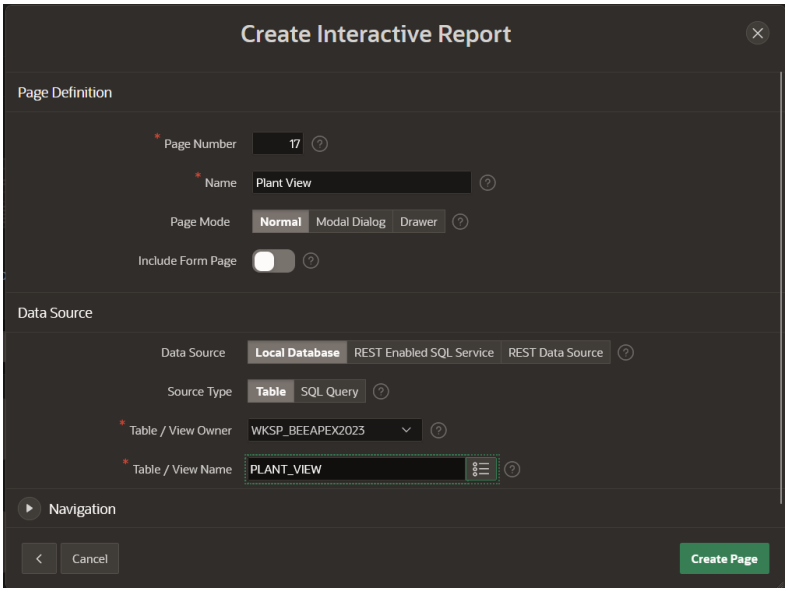
The simplest way to display multiple tables on one form is to create view in *SQL Workshop*, and then create interactive report that uses that view. *Query Builder* can be used to graphically define a view. The steps with captured screens are shown below:

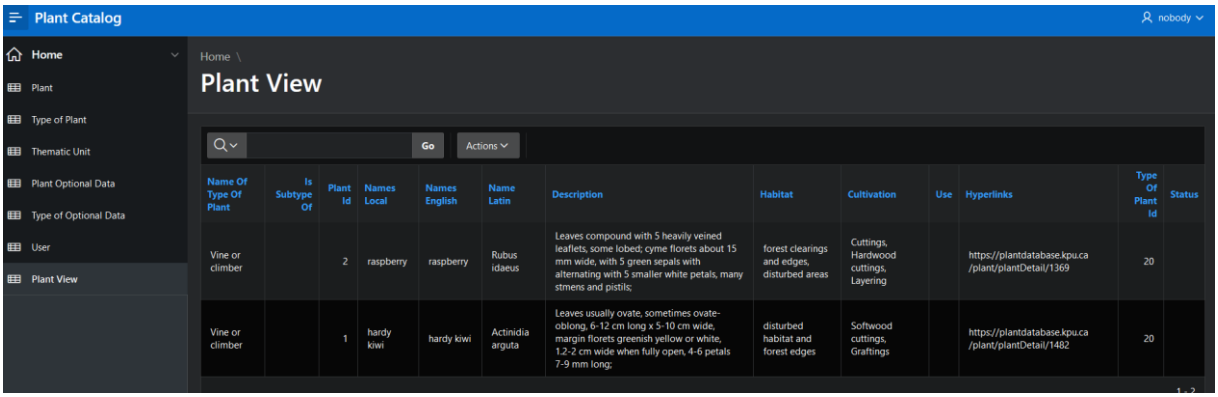





GREENDI_USER
PLANT
PLANT_OPTIONAL_DATA
PLANT_THEMATIC_UNIT
THEMATIC_UNIT
TYPE_OF_OPTIONAL_DATA
TYPE_OF_PLANT
TYPE_OF_PLANT
TYPE_OF_PLANT_ID
NAME_OF_TYPE_OF_PLANT
IS_SUBTYPE_OF
PLANT
HABITAT
CULTIVATION
USE
PHOTO
HYPERLINKS
STATUS
TYPE_OF_PLANT_ID

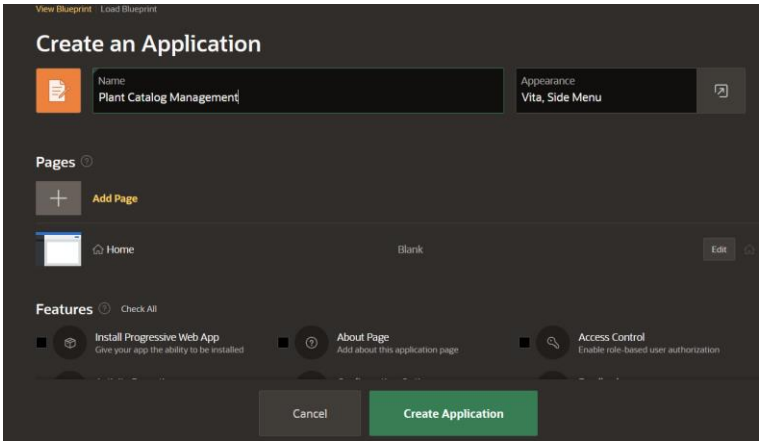

Schema
WKSP_BEEAPEX2023
View Name
plant_view
Preserve Case
Query
select TYPE_OF_PLANT.NAME_OF_TYPE_OF_PLANT as NAME_OF_TYPE_OF_PLANT,
TYPE_OF_PLANT.TYPE_OF_PLANT_ID as TYPE_OF_PLANT_ID,
TYPE_OF_PLANT.IS_SUBTYPE_OF as IS_SUBTYPE_OF,
PLANT.PLANT_ID as PLANT_ID,
PLANT.NAMES_LOCAL as NAMES_LOCAL,
PLANT.NAMES_ENGLISH as NAMES_ENGLISH,
PLANT.NAME_LATIN as NAME_LATIN,
PLANT.DESCRIPTION as DESCRIPTION,
PLANT.HABITAT as HABITAT,
PLANT.CULTIVATION as CULTIVATION,
PLANT.USE as USE,
PLANT.HYPERLINKS as HYPERLINKS,
PLANT.TYPE_OF_PLANT_ID as TYPE_OF_PLANT_ID,
Query Builder
SQL Commands
A view is a tailored presentation of the data contained in one or more tables or other views. A view takes the output of a query and treats it as a table. Think of a view as a stored query or a virtual table.
Cancel
Next


Create Interactive Report
Page Definition
Page Number
17
Name
Plant View
Page Mode
Normal
Modal Dialog
Drawer
Include Form Page
Data Source
Data Source
Local Database
REST Enabled SQL Service
REST Data Source
Source Type
Table
SQL Query
Table / View Owner
WKSP_BEEAPEX2023
Table / View Name
PLANT_VIEW
Navigation
Cancel
Create Page

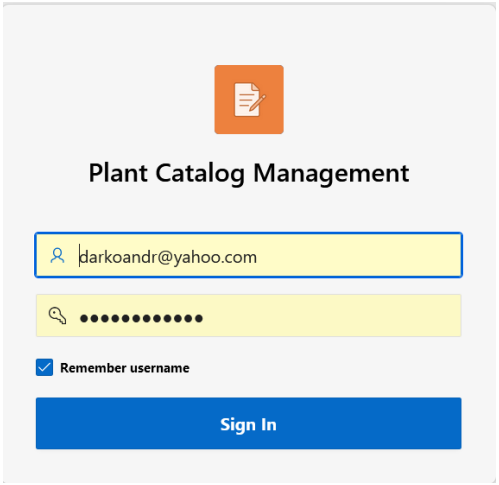

Plant Catalog
nobody
Home
Plant
Type of Plant
Thematic Unit
Plant Optional Data
Type of Optional Data
User
Plant View
Home
Plant View
Go
Actions
Name Of Type Of Plant
Is Subtype Of
Plant Id
Names Local
Names English
Name Latin
Description
Habitat
Cultivation
Use
Hyperlinks
Type Of Plant Id
Status
Vine or climber
2
raspberry
raspberry
Rubus idaeus
Leaves compound with 5 heavily veined leaflets, some lobed; cyme florets about 15 mm wide, with 5 green sepals with alternating with 5 smaller white petals, many stmens and pistils;
forest clearings and edges, disturbed areas
Cuttings, Hardwood cuttings, Layering
https://plantdatabase.kpu.ca/plant/plantDetail/1369
20
Vine or climber
1
hardy kiwi
hardy kiwi
Actinidia arguta
Leaves usually ovate, sometimes ovate-oblong, 6-12 cm long x 5-10 cm wide, margin florets greenish yellow or white, 1.2-2 cm wide when fully open, 4-6 petals 7-9 mm long;
disturbed habitat and forest edges
Softwood cuttings, Graftings
https://plantdatabase.kpu.ca/plant/plantDetail/1482
20

Test application link (public part – open browsing of plant data):
https://apex.oracle.com/pls/apex/r/beeapex2023/plant-catalog/plant

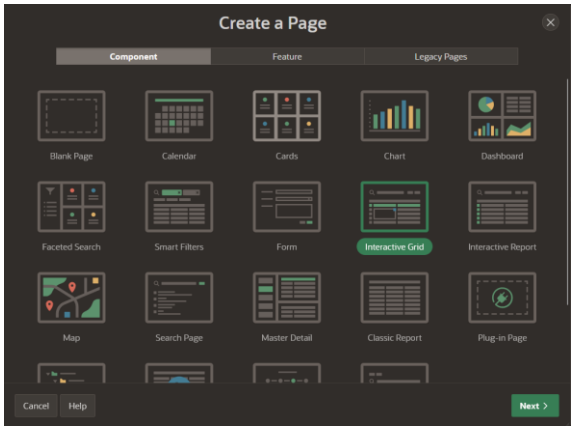
Creating an Oracle Apex application – private (administrative) part – Plant Catalog Management

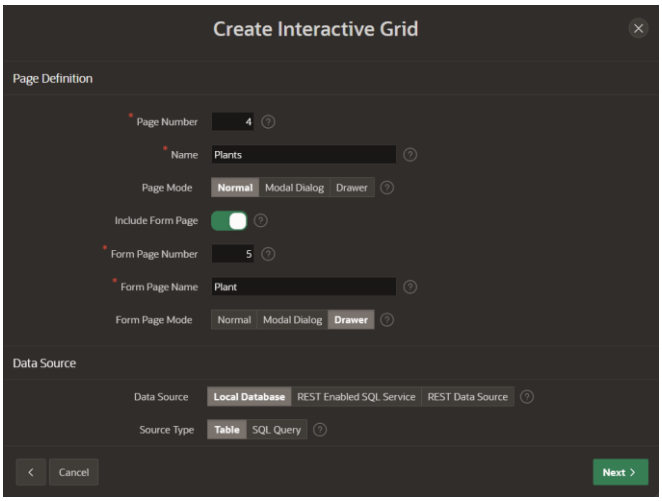


Login page:

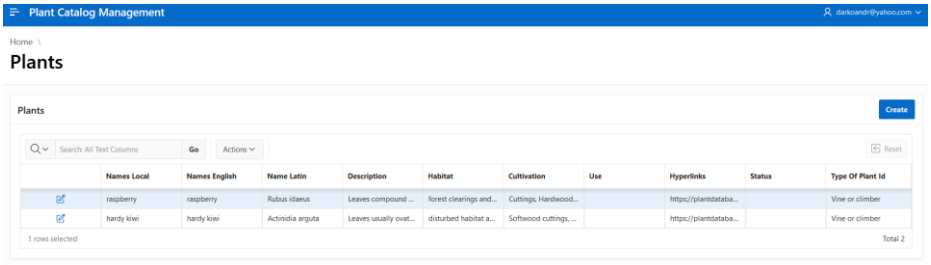


Next, we will create *Interactive Grid* type of pages with included Form Page to be able to create, update, delete and insert data into all of the tables from our data model:

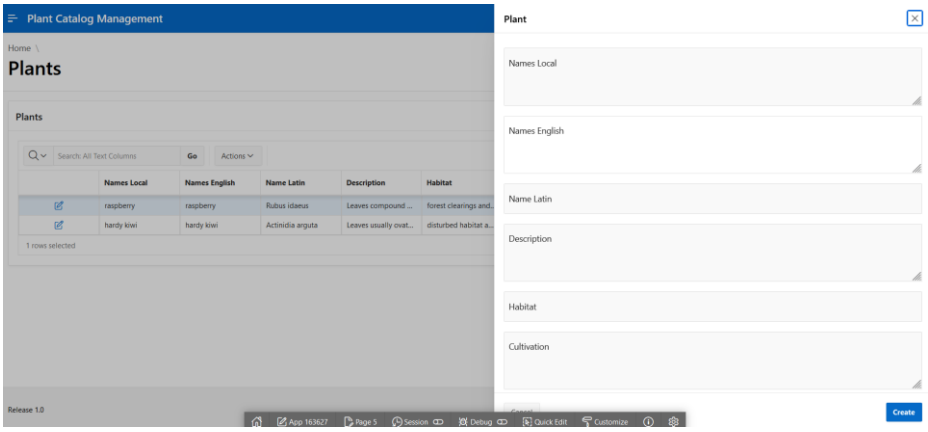




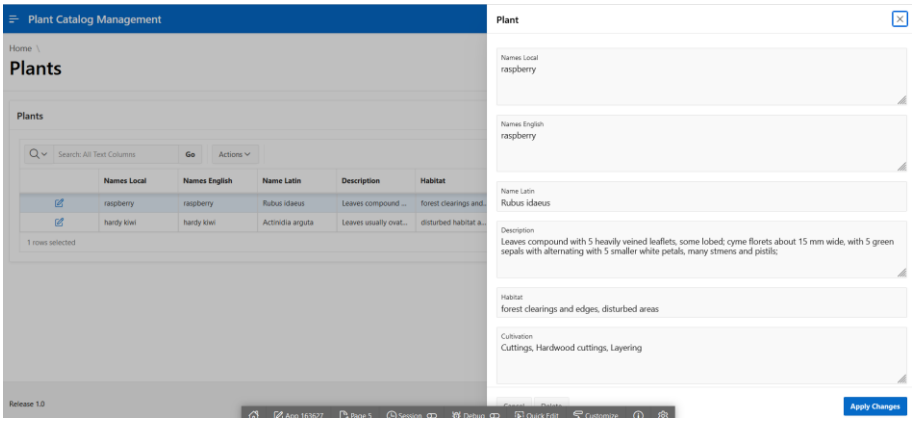
Main page of interactive grid for Plant table:



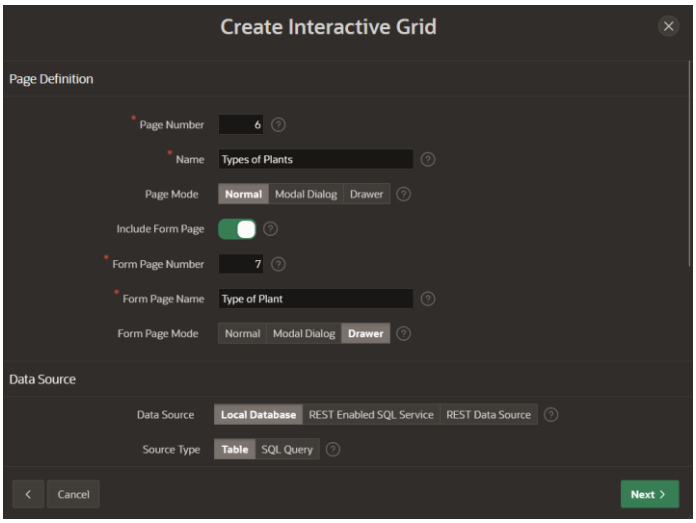
Create page of interactive grid for Plant table:

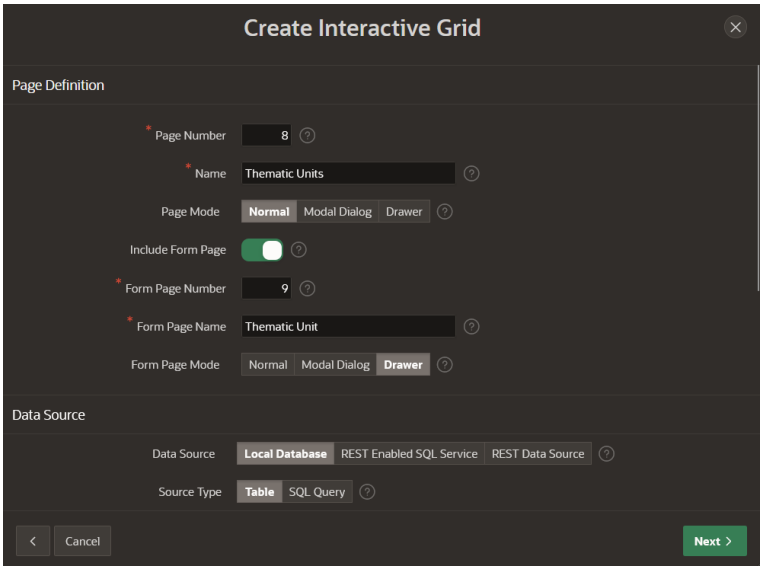


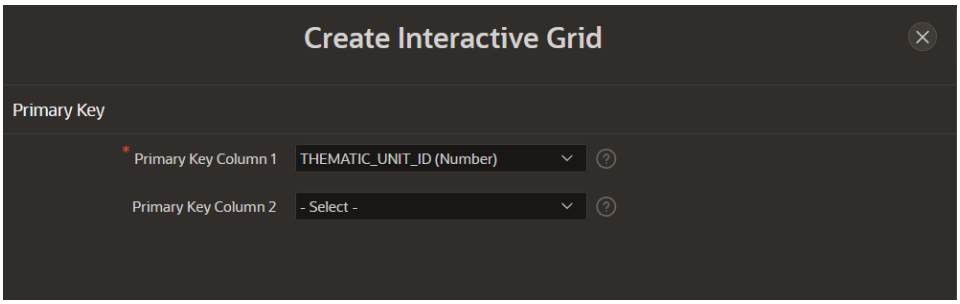
Edit and delete page of interactive grid for Plant table:

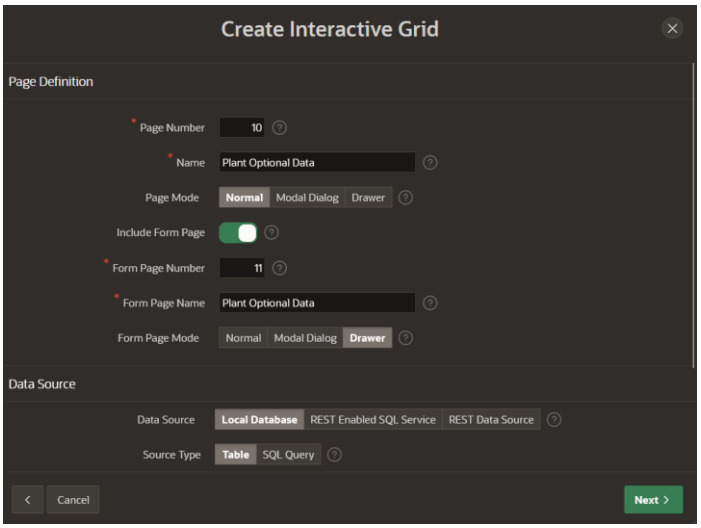


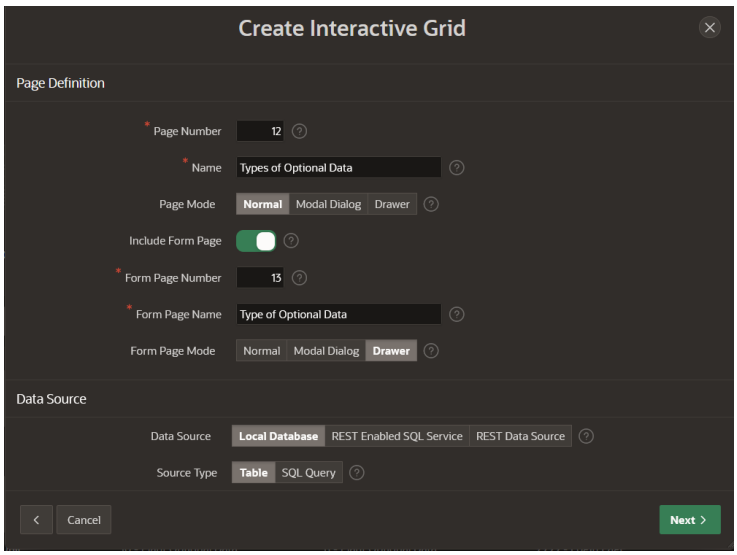
The same can be done for all other tables (*type_of_plant*, *thematic_unit*, *plant_optional_data*, *type_of_optional_data*, *user*):

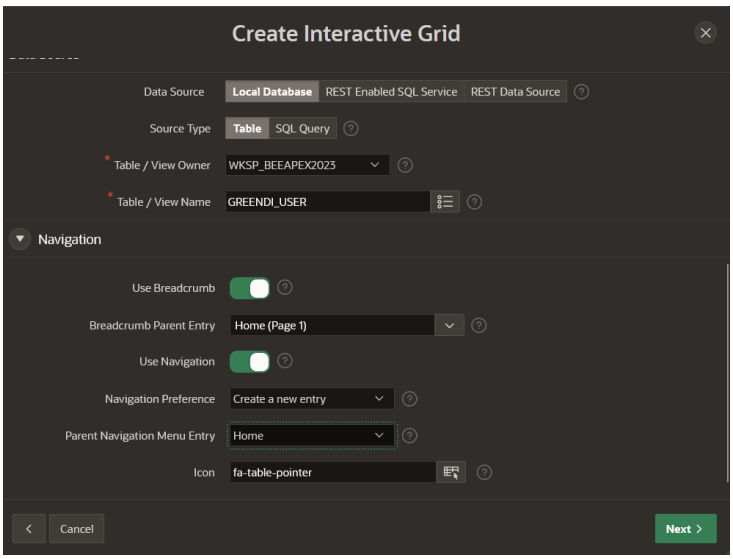












Test application link - private (administrative) part – Plant Catalog Management:

https://apex.oracle.com/pls/apex/r/beeapex2023/plant-catalog-management/plants