

ORACLE®

Oracle APEX Hands-On Lab

Building a Proof-of-Concept
for Oracle Autonomous Cloud Service

July, 2019 (v19.1.3)

Step Up to
Modern Cloud
Development



Overview

This lab starts with a napkin design of data structures we would like to use for a proof-of-concept. Using Quick SQL you will quickly define the data structures and utilize various table and column directives to better define the tables and also create some dummy data. You will then be creating an application based on the new tables. Lastly, you will be updating some of the generated components to improve the initial app.

From napkin design to demo-ready, proof-of-concept app, complete with some dummy data, in minutes - Viola!

Please note this lab assumes that you already have the Autonomous Data Warehouse (ADW) or Autonomous Transaction Processing – Serverless (ATP-S) service provisioned.

If you do not currently have a cloud service, sign up for a free trial account at <https://cloud.oracle.com/try-autonomous-database>

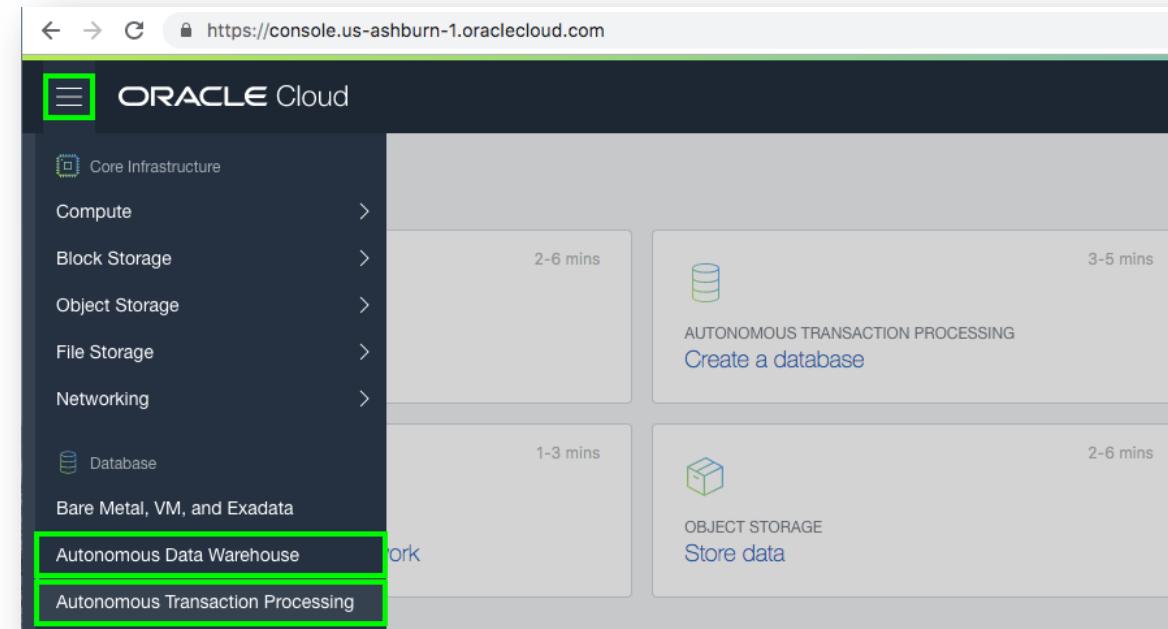
Getting Started

Obtaining a Workspace

{Note: If you have a workspace on the Autonomous Database Cloud Service then you can skip this section and move to Section 2}

Step 1.1 – Accessing APEX

- Sign into your Oracle Cloud service
- Click the hamburger (top left), select Autonomous Data Warehouse or Autonomous Transaction Processing, based on which service(s) you have defined.



Step 1.1b – Accessing APEX

- Click <Your Database> from the list

The screenshot shows the Oracle Cloud interface with the title "Autonomous Databases in dpeakea (root)". On the left, there's a sidebar with options: "Autonomous Database" (selected), "Autonomous Container Database", and "Autonomous Exadata Infrastructure". On the right, a table lists databases. One row is highlighted with a green border: "Name": "dpeake", "Database Name": "dpeake", "State": "Available" (green dot icon), and "Dedicated Infrastructure": "No".

Name	Database Name	State	Dedicated Infrastructure
dpeake	dpeake	Available	No

- Click Service Console

The screenshot shows the "Autonomous Database Details" page for the database "dpeake". At the top, there's a large green button with "ATP" in white. Below it, the word "AVAILABLE" is displayed. To the right, there are several buttons: "DB Connection", "Service Console" (which is highlighted with a green border), "Scale Up/Down", "Stop", and "Actions". Underneath these buttons, there are tabs: "Autonomous Database Information" (selected) and "Tags". In the "General Information" section, it says "Database Name: dpeake" and "Workload Type: Transaction Processing".

Step 1.1c – Accessing APEX

- Click Development
- Click APEX

The screenshot shows the Oracle Cloud Infrastructure homepage. On the left, there's a sidebar with navigation links: Autonomous, Transaction Processing, Overview, Activity, Administration, and Development. The Development link is highlighted with a green box. Below the sidebar, there are several cards. One card for 'APEX' is highlighted with a green box. The APEX card contains the following text: 'APEX provides a low-code development environment that enables you to build apps in a single, extensible platform, which is fully supported by Autonomous Database.' Another card for 'SQL Developer Web' is partially visible. At the bottom of the page, there's a red footer bar with the Oracle logo.

Autonomous
Transaction Processing

Overview
Activity
Administration
Development
DATABASE
DPEAKE

APEX

APEX provides a low-code development environment that enables you to build apps in a single, extensible platform, which is fully supported by Autonomous Database.

SQL Developer Web

Oracle SQL Developer Web provides a browser-based integrated development environment and administration interface for Oracle Autonomous Database. It provides a subset of the features available in the desktop product.

OML Notebooks

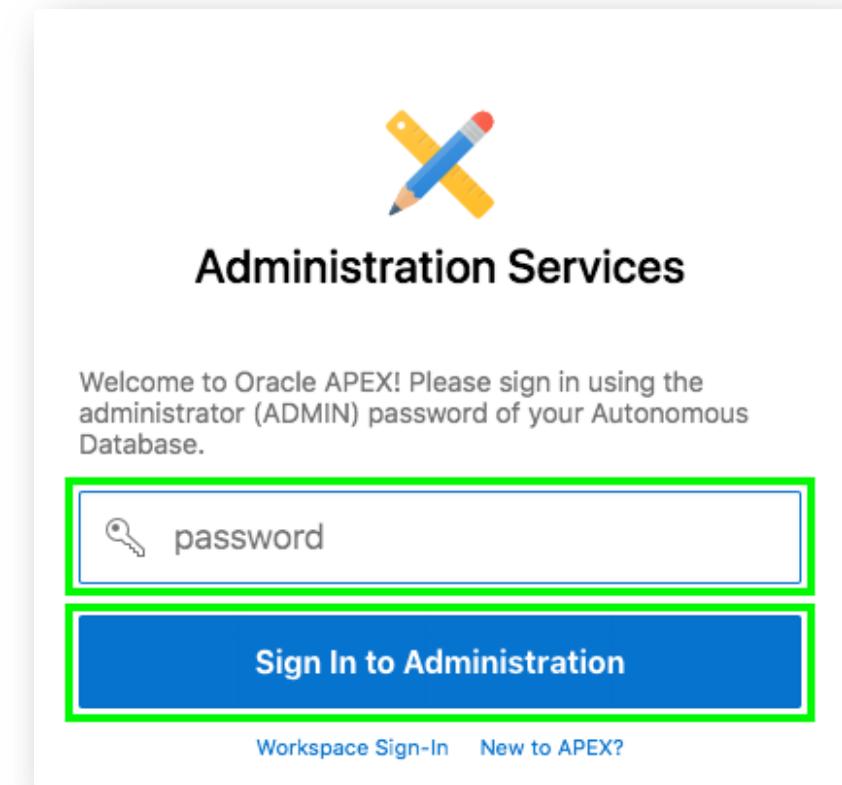
OML SQL notebooks provide easy access to Oracle's parallelized, scalable in-database implementations of a library of Oracle Advanced Analytics' machine learning algorithms (classification, regression, anomaly detection, clustering, associations, attribute importance, feature extraction, times series, etc.), SQL, PL/SQL and Oracle's statistical and analytical SQL functions.

Download Oracle Instant Client

This is a free, light-weight set of tools, libraries and SDKs for building and connecting applications. These libraries underly the Oracle APIs of languages including Node.js, Python and PHP and provide access for OCI, OCCI, JDBC, ODBC and Pro*C applications. Tools such as SQL*Plus and Oracle Data Pump are also included - Oracle recommends using this version of Data Pump for moving existing Oracle Database schemas to Autonomous Transaction Processing.

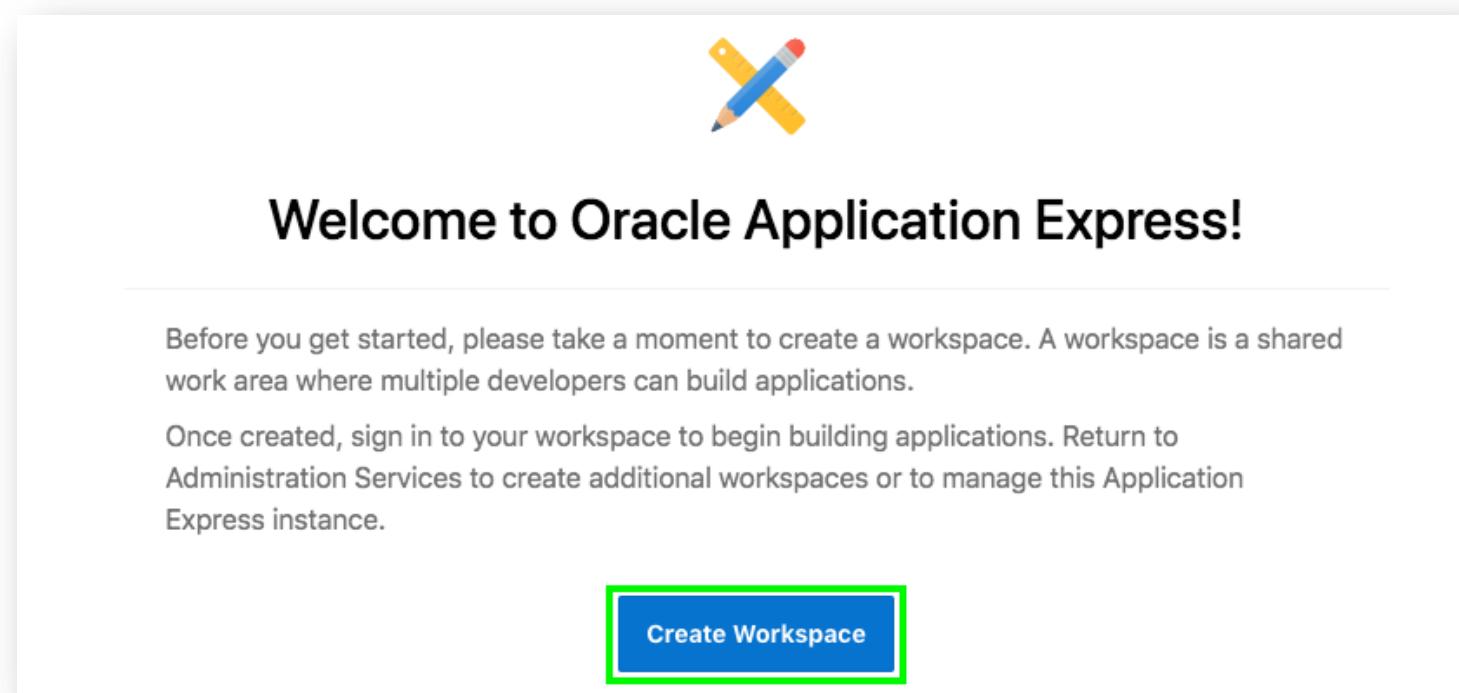
Step 1.2 – Creating a Workspace

- To sign into APEX Administrative Services for Password enter your OCI Password
- Click **Sign In to Administration**



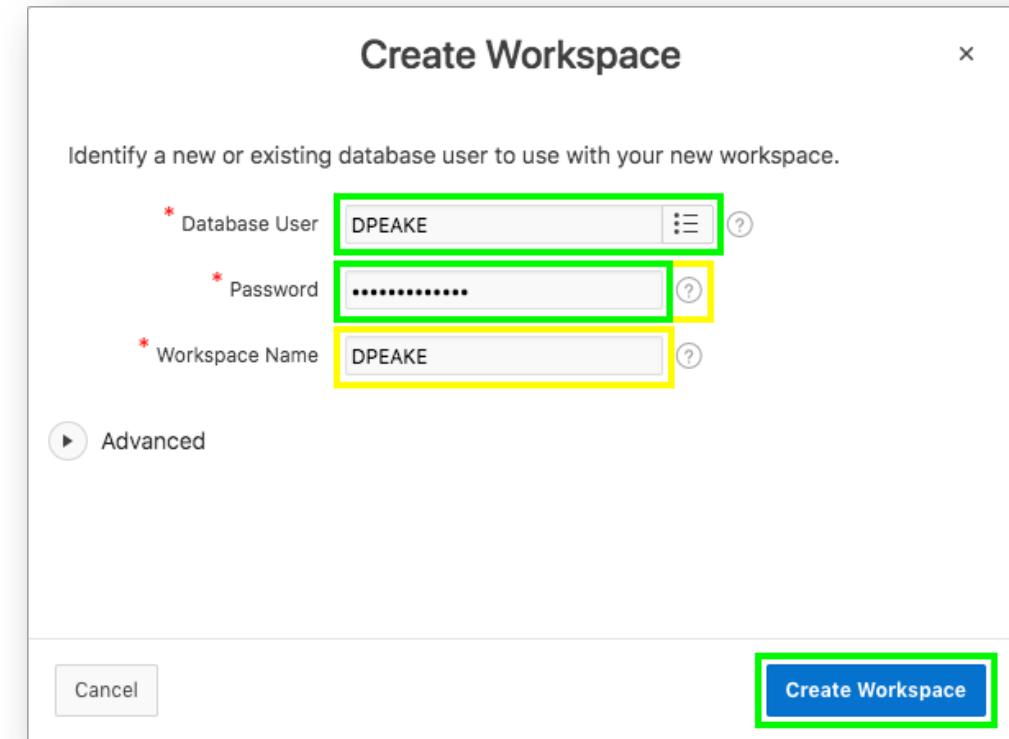
Step 1.2b – Creating a Workspace

- Given this is your first time entering APEX, click **Create Workspace**



Step 1.2c – Creating a Workspace

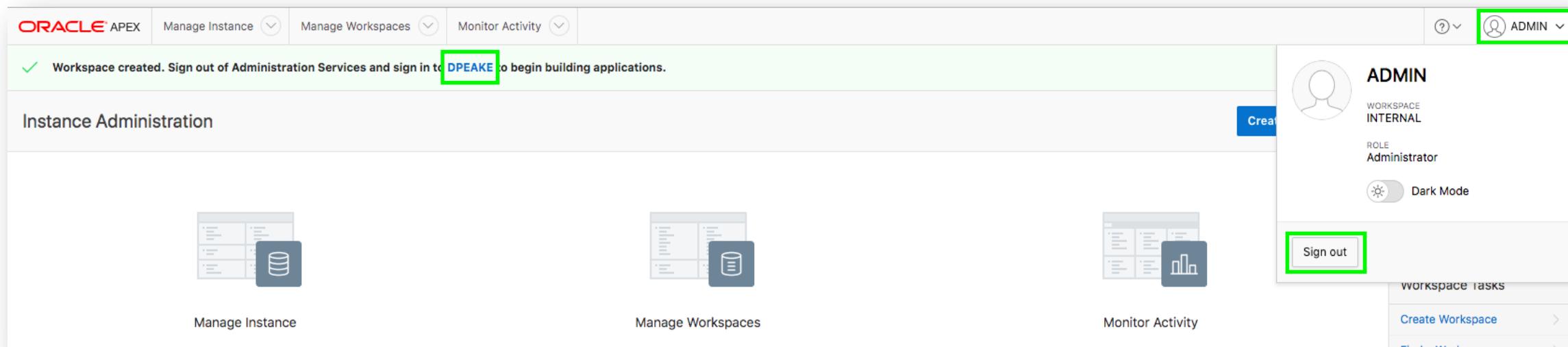
- For Database User enter an appropriate name
- Enter a Password {Click the ? Icon to see password complexity rules}
- Click **Create Workspace**



{Note: The Database User will be used for the Workspace Name. If you want you can update the Workspace Name}

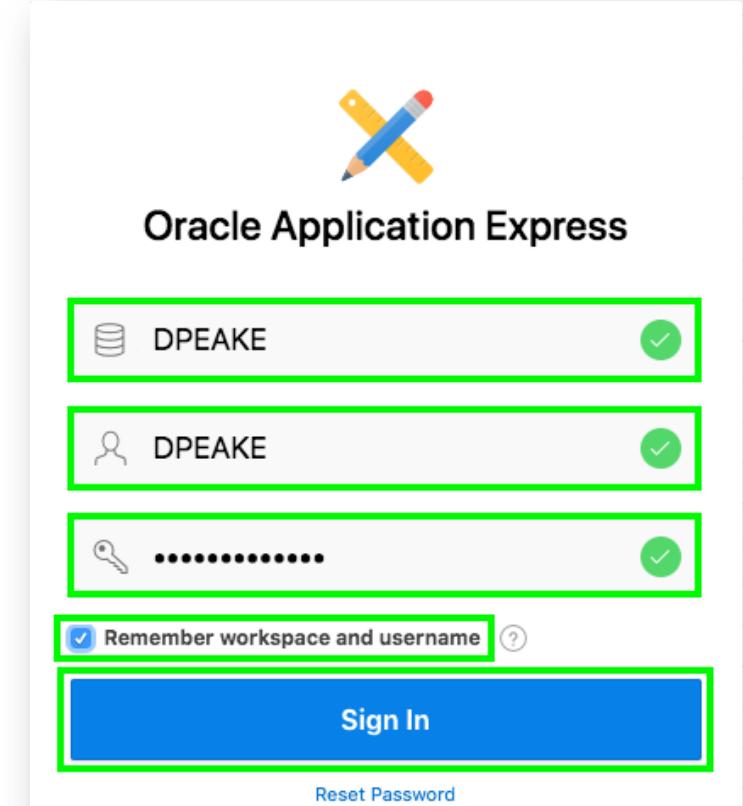
Step 1.3 – Log into your New Workspace

- Click on the link within the success message {easiest technique}
OR
Click on the Admin user (top right), click **Sign Out**,
and then click **Return to Sign In Page**



Step 1.3b – Log into your New Workspace

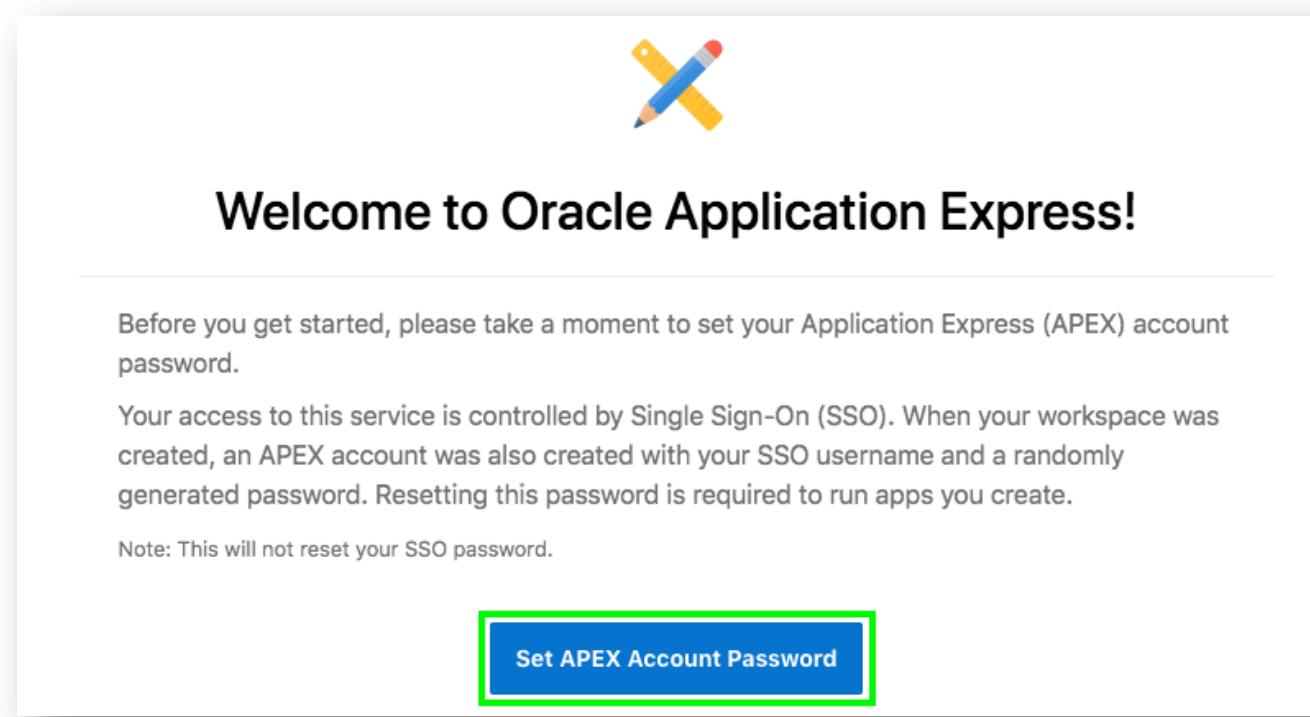
- Sign into your new Workspace
Workspace – enter **<Your Workspace Name>**
Username – enter **<Your Database User>**
Password – enter your OCI Password
Remember workspace and username - Check
- Click **Sign In**
- *{Note: Enter the Workspace Name and Database User entered in Step 2c above}*



The image shows the Oracle Application Express login page. At the top right is a yellow pencil icon with a red 'X' through it. Below it is the text "Oracle Application Express". The page contains four input fields, each with a green border and a checkmark in a green circle. The first field has a database icon and the text "DPEAKE". The second field has a user icon and the text "DPEAKE". The third field has a key icon and a series of dots representing a password. Below these fields is a checkbox labeled "Remember workspace and username" with a question mark icon next to it. At the bottom is a large blue "Sign In" button and a smaller "Reset Password" link.

Step 1.3c – Log into your New Workspace

- Given this is your first time entering your new Workspace, click **Set APEX Account Password**



Step 1.3d – Log into your New Workspace

- For your user profile enter the following:
Email Address – enter your email address
Enter New Password – enter your OCI Password
Confirm Password – enter your OCI Password
- Click **Apply Changes**

The screenshot shows the 'Edit Profile' dialog box. In the 'Profile Details' section, the 'Email Address' field contains 'david.peake@oracle.com'. In the 'Password' section, both the 'Enter New Password' and 'Confirm Password' fields contain masked password entries. The 'Apply Changes' button at the bottom right is highlighted with a large blue box.

Defining new data structures

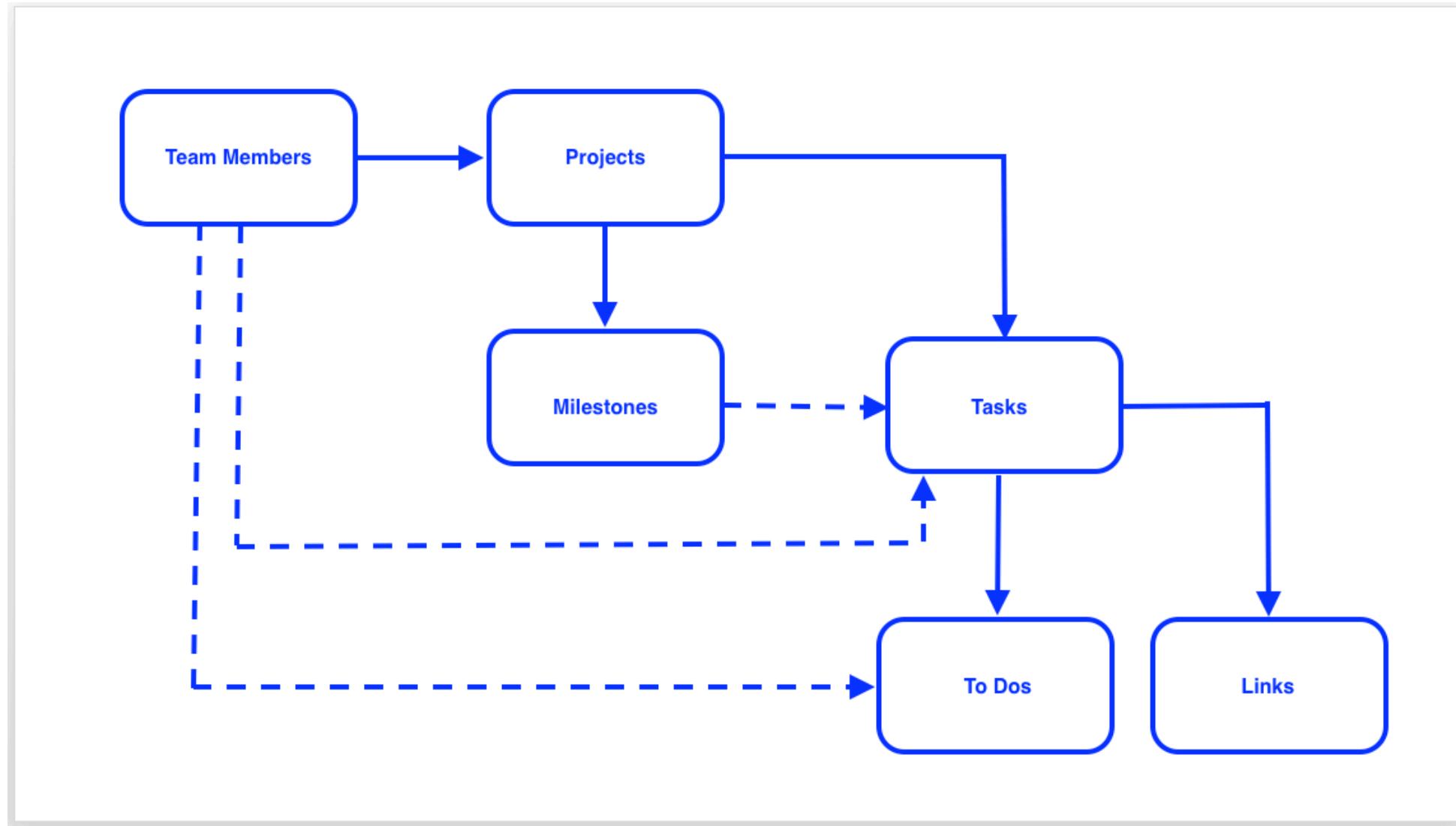
Using Quick SQL

Original Spreadsheet Table

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. The left sidebar displays the Object Browser with a 'Tables' dropdown and a search bar. The main area is titled 'SPREADSHEET' and contains a table definition for a new table.

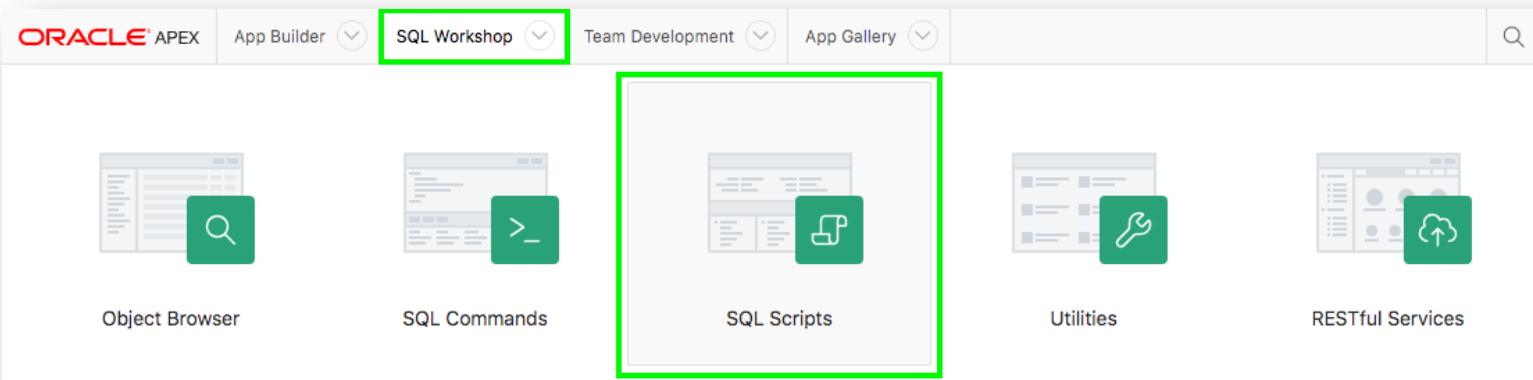
Column Name	Data Type	Nullable
ID	NUMBER	No
PROJECT	VARCHAR2(30)	Yes
TASK_NAME	VARCHAR2(255)	Yes
START_DATE	DATE	Yes
END_DATE	DATE	Yes
STATUS	VARCHAR2(30)	Yes
ASSIGNED_TO	VARCHAR2(30)	Yes
COST	NUMBER	Yes
BUDGET	NUMBER	Yes

Napkin Design – Improved data model for Projects

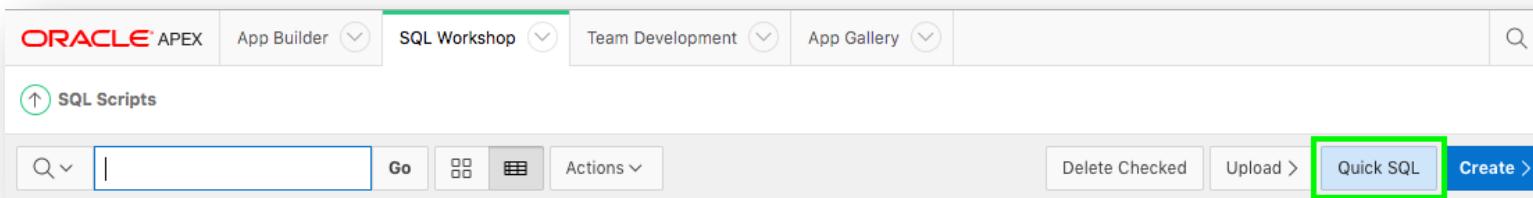


Step 2.1 – Open Quick SQL

- Log into your workspace
- Click **SQL Workshop**
- Click **SQL Scripts**



- Click **Quick SQL**



Step 2.2 - Enter shorthand for Team Members table

- Enter the Table Name {Team Members}
- Indent 2 or more spaces and enter the column names

The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. In the 'Quick SQL Shorthand' section, the user has entered:

```
1 Team Members
2 username
3 full name
4 email
5 phone_number
6 profile
7 photo file
8
```

A green box highlights the first line '1 Team Members'. In the 'Oracle SQL Output' section, the generated SQL script is displayed:

```
1 -- create tables
2 create table team_members (
3   id
4   username
5   full_name
6   email
7   phone_number
8   profile
9   photo
10  photo_filename
11  photo_mimetype
12  photo_charset
13  photo_lastupd
14 )
15;
```

A purple box highlights the entire generated SQL script.

{Note: You don't need to enter all of the column names shown, as you will load a complete script later in this lesson}

Step 2.3 - Enter details for Projects table

- Enter the Table Name in the first column {Projects}
- Indent 2 or more spaces and enter the column names

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes tabs for ORACLE APEX, App Builder, SQL Workshop (selected), Team Development, and Packaged Apps. The Utilities \ Quick SQL section is active. The left pane displays SQL shorthand code for creating tables:

```
1 Team Members
2   username
3   full name
4   email
5   phone_number
6   profile
7   photo file
8 Projects
9   name
10  project lead
11  budget
12  status
13  completed_date
14  description
```

The right pane shows the generated Oracle SQL Output:

```
-- create tables
create table team_members (
  id          number not null constraint team_members_id_pk primary key,
  username    varchar2(255),
  full_name   varchar2(255),
  email       varchar2(255),
  phone_number number,
  profile     varchar2(4000),
  photo       blob,
  photo_filename varchar2(512),
  photo_mimetype varchar2(512),
  photo_charset  varchar2(512),
  photo_lastupd date
);

create table projects (
  id          number not null constraint projects_id_pk primary key,
  name        varchar2(255),
  project_lead varchar2(4000),
  budget      varchar2(4000),
  status      varchar2(60),
  completed_date date
);
```

Both the shorthand code and the generated SQL output are highlighted with green boxes.

Step 2.4 – Review Help

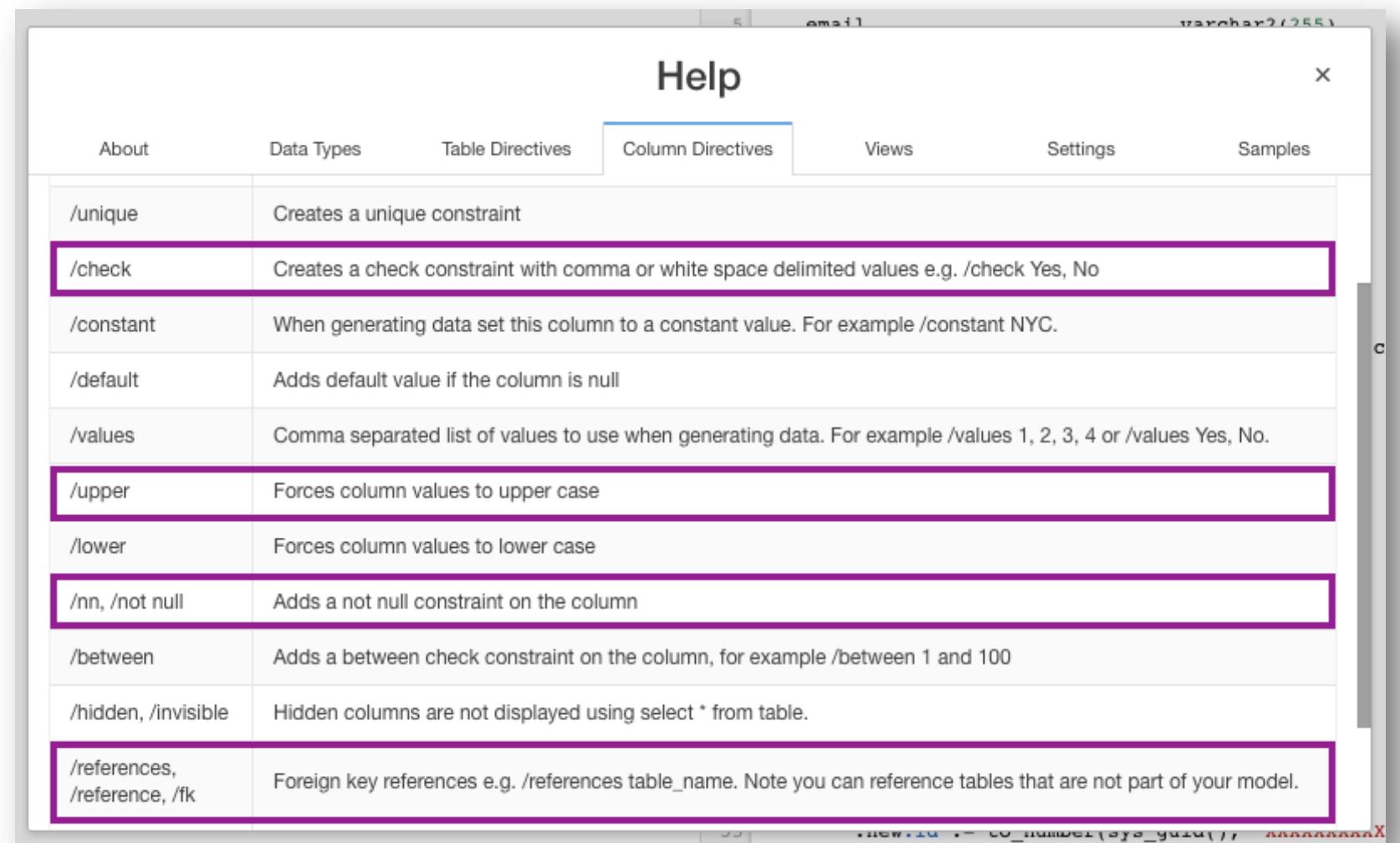
- Click Help
- Click Table Directives

The screenshot shows the Oracle SQL Developer interface. On the left, there's a 'Quick SQL Shorthand' panel with a list of abbreviations for various database objects. In the center, there's a 'Help' panel with tabs for 'About', 'Data Types', 'Table Directives' (which is highlighted with a green box), 'Column Directives', 'Views', 'Settings', and 'Samples'. Below these tabs is a section titled 'Table Directives' containing a table of directives:

Directive	Description
/api	Generate PL/SQL package API to query, insert, update, and delete data within a table.
/audit	Adds Oracle auditing, by default AUDIT ALL ON [TABLE NAME].
/auditcols, /audit cols, /audit columns	Automatically adds an UPDATED, UPDATED_BY, INSERTED, and INSERTED_BY columns and the trigger logic to set column values.
/colprefix	Prefix all columns of a given table with this value. Automatically adds an underscore if not provided.
/compress, /compressed	Table will be created compressed.
/history	Generate code to log changes into a history table for this specific table.
/insert NN	Generate NN SQL INSERT statement(s) with random data, for example: /INSERT 20. (Maximum = 1000)
/rest	Generate REST enablement of the table using Oracle REST Data Services (ORDS)
/select	Generate SQL SELECT statement after generating data for each table

Step 2.4b – Review Help

- Click Column Directives



Step 2.4c – Review Help

- Click Data Types

The screenshot shows the Oracle Database SQL Developer Help window. The title bar says "Help". The navigation bar includes "About", "Data Types" (which is highlighted with a blue border), "Table Directives", "Column Directives", "Views", "Settings", and "Samples". The main content area is titled "Data Types" and contains a table of data types:

num, number	NUMBER
int, integer	INTEGER
d, date	DATE
ts, timestamp	TIMESTAMP
tstz, tswtz, timestamp with local time zone	TIMESTAMP WITH LOCAL TIMEZONE
char, vc, varchar, varchar2, string	VARCHAR2(4000)
vcNNN	VARCHAR2(NNN) NNN identifies a number between 1 and 32767.
vc(NNN)	VARCHAR2(NNN) NNN identifies a number between 1 and 32767.
vc32k	VARCHAR2(32767)

Step 2.5 - Improve the Shorthand

- Close Help
- Enter **/insert xx** for tables
- Enter **/nn** for mandatory columns
- Enter **/references team_members** for project lead column
- Enter **num** for budget column
- Enter **/vc30** and **/check ASSIGNED, IN-PROGRESS, COMPLETED** for status column

```
1 Team Members /insert 10
2   username /nn /upper
3   full name
4   email /nn
5   phone_number
6   profile
7   photo
8 Projects /insert 20
9   name /nn
10  project lead /nn /references team_members
11  budget num
12  status vc30 /nn /check ASSIGNED, IN-PROGRESS,
13  completed_date
14  description
15
```

Step 2.6 - Enter details for a child table

- Enter the Table Name indented {Milestones}
- Indent 2 or more spaces and enter the column names

```
31 create table milestones (
32     id                      number not null constraint milestones_id_pk pr
33     project_id               number
34                                         constraint milestones_project_id_fk
35                                         references projects on delete cascade,
36     name                     varchar2(255) not null,
37     due_date                 date not null,
38     description              varchar2(4000)
39 )
```

```
1 Team Members /insert 10
2     username /nn /upper
3     full name
4     email /nn
5     phone_number
6     profile
7     photo
8 Projects /insert 20
9     name /nn
10    project lead /nn /refer
11    budget num
12    status vc30 /nn /check .
13    completed_date
14    description
15 Milestones /insert 30
16     name /nn
17     due_date /nn
18     description
19
```

Step 2.7 - Enter details for another child table

- Enter the Table Name indented {Tasks}
- Indent 2 or more spaces and enter the column names

```
1 Team Members /insert 10
2   username /nn /upper
3   full name
4   email /nn
5   phone_number
6   profile
7   photo
8 Projects /insert 20
9   name /nn
10  project lead /nn /references team_members
11  budget num
12  status vc30 /nn /check ASSIGNED, IN-PROGR
13  completed_date
14  description
15 Milestones /insert 30
16  name /nn
17  due_date /nn
18  description
19 Tasks /insert 100
20  name /nn
21  assignee /nn /references team_members
22  milestone_id /references milestones
23  start_date /nn
24  end_date
25  cost num
26  description
27  is_complete_yn /check Y, N
```

Step 2.8 – Complete the Shorthand

- Copy the following URL into a new window in your browser:
{Remember you are on Slide 21 if you click the link directly}

<http://www.oracle.com/technetwork/developer-tools/apex/application-express/apex-beginner-quicksql-5095785.txt>

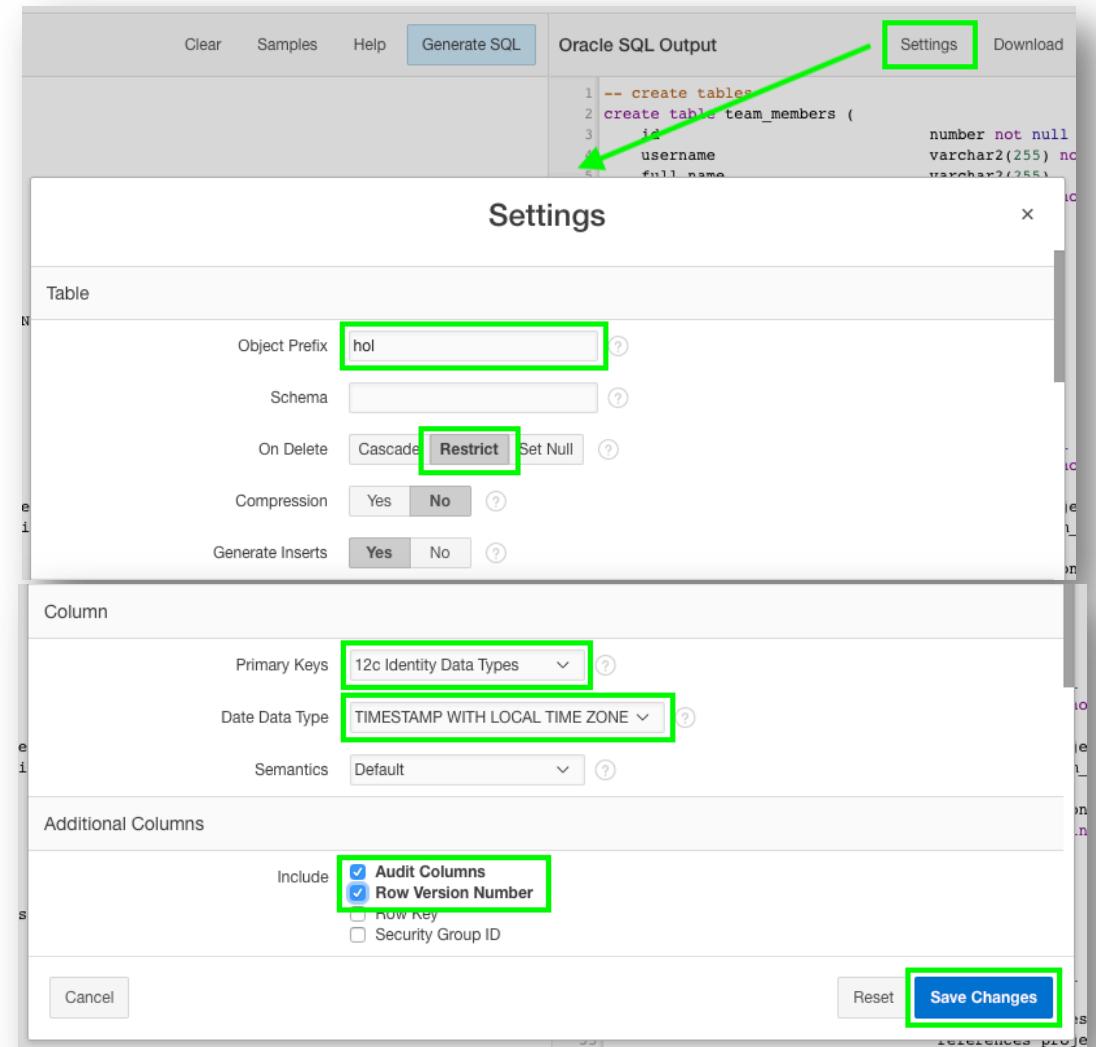
- Copy and Paste the full script
into the Quick SQL Shorthand pane
{on the left}
- Click **Generate SQL**

```
team_members /insert 10
username /nn /upper
full name
email /nn
phone_number
profile
photo file
projects /insert 20
name /nn
project_lead /nn /references team_members
budget num
status vc30 /nn /check ASSIGNED, IN-PROGRESS, COMPLETED
completed_date
description
milestones /insert 30
name /nn
due_date /nn
description
tasks /insert 100
name /nn
assignee /nn /references team_members
milestone_id /references milestones
start_date /nn
end_date
cost num
description
is_complete_yn /check Y, N

view project_tasks projects tasks
```

Step 2.9 – Update the Settings

- Click **Settings**
- Object Prefix, enter **hol**
- On Delete, select **Restrict**
- Primary Keys, select **12c Identity Data Types**
- Date Data Type, select **TIMESTAMP WITH LOCAL TIME ZONE**
- Audit Columns, check **Include**
- Row Version Number, check **Include**
- Click **Save Changes**

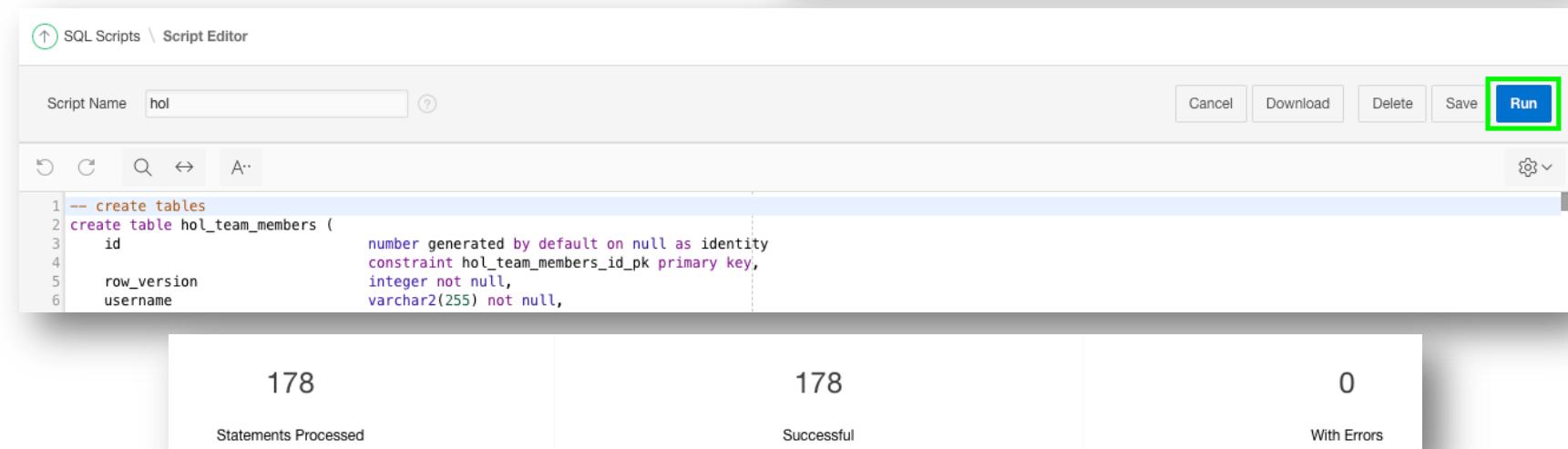
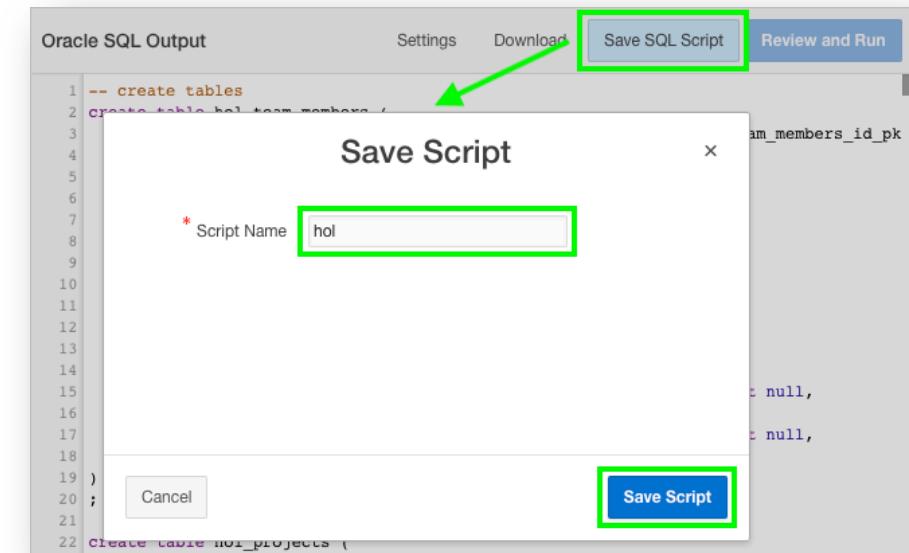


Step 2.10 – Save, Review, and Run the Script

- Click Save SQL Script
- For Script Name, enter hol
- Click Save Script
- Click Review and Run

{Note: The script will be displayed in the Script Editor within SQL Scripts}

- Click Run
- Click Run Now



Creating an app on the tables from Quick SQL

Using the Create Application Wizard

Step 3.1 – Start the Create App Wizard

- Click Create App from Script

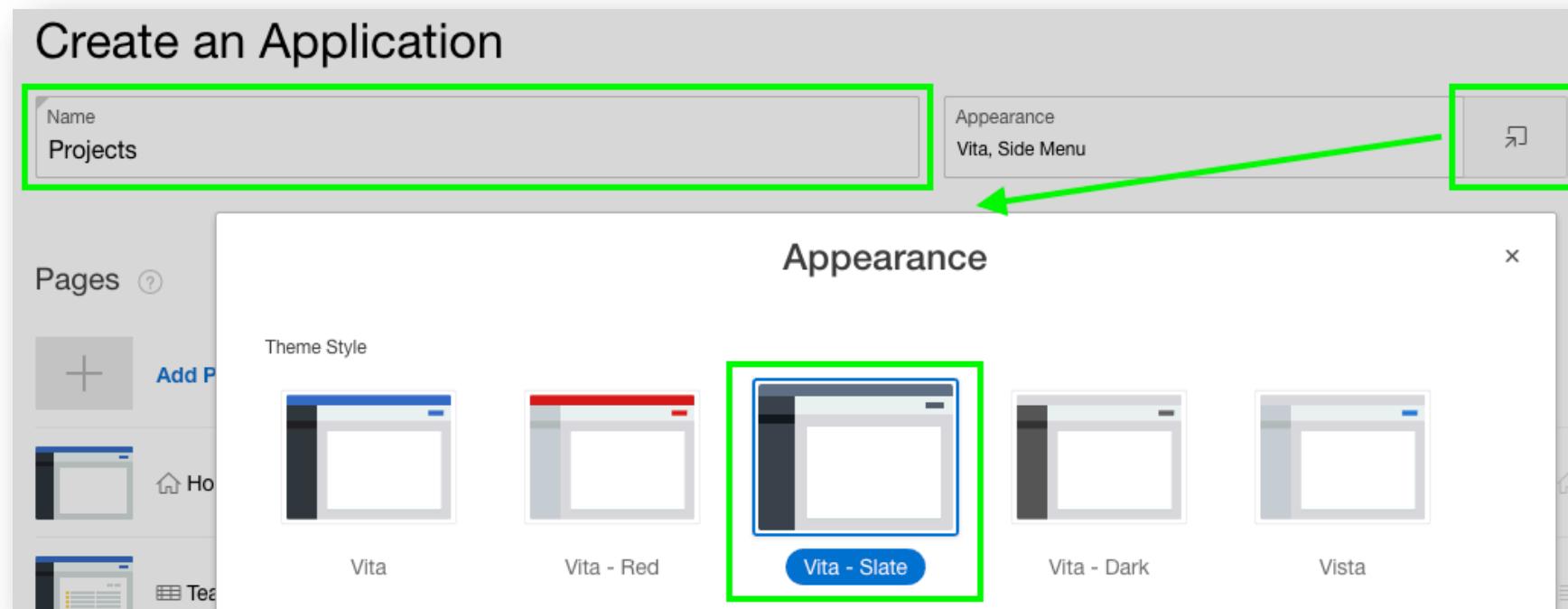
The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. The results of a script named 'hol' are displayed, showing one statement: 'create table hol_team_members (id'. The status is 'Complete'. The 'Create App from Script' button is highlighted with a green box.

Number	Elapsed	Statement	Feedback	Rows
1	0.06	create table hol_team_members (id	Table created.	0

{Note: If you are back on SQL Scripts, and don't see the “Create App from Script” button perform the following steps:
1. Within the Results column, click “1” for the script you just ran.
2. Under View Results, click the magnifying glass.
The results page shown above should now be displayed again}

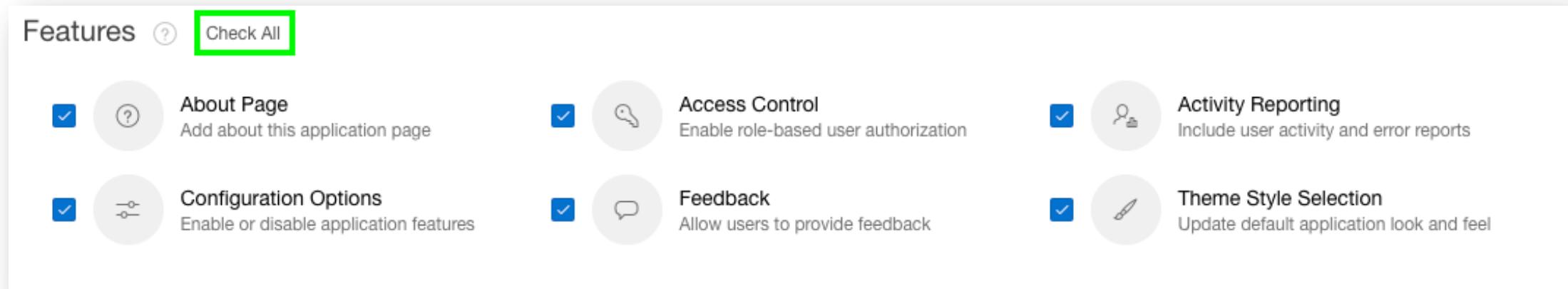
Step 3.2 – Name the App and Update Appearance

- For Name, enter Projects
- Click Appearance
- For Theme Style, select Vita-Slate



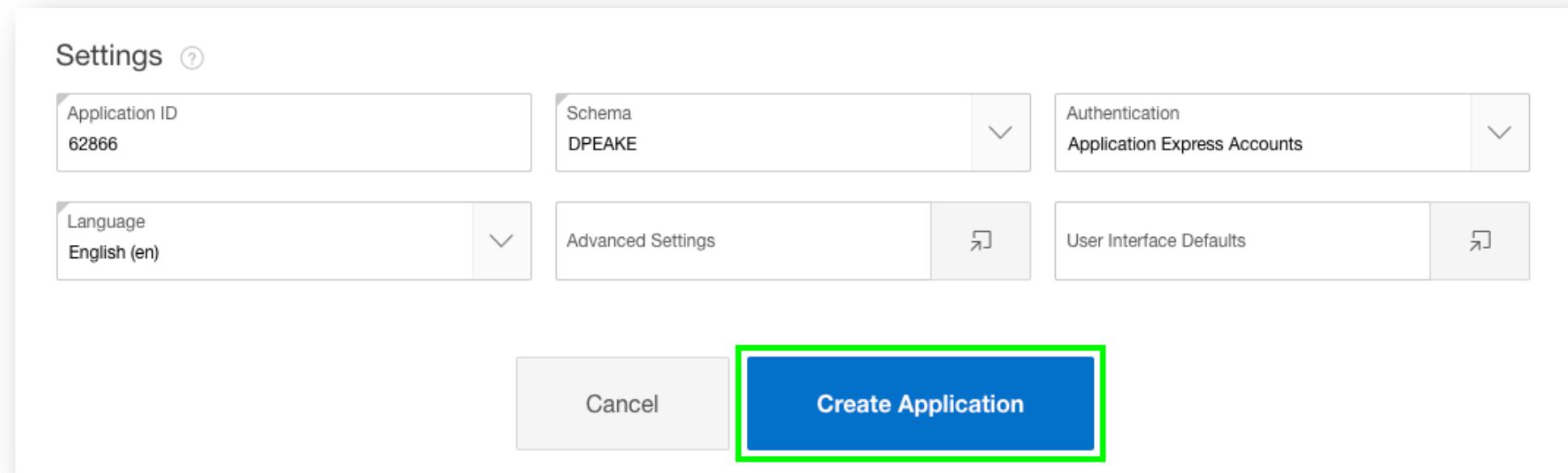
Step 3.3 – Add Features

- For Features, click Check All



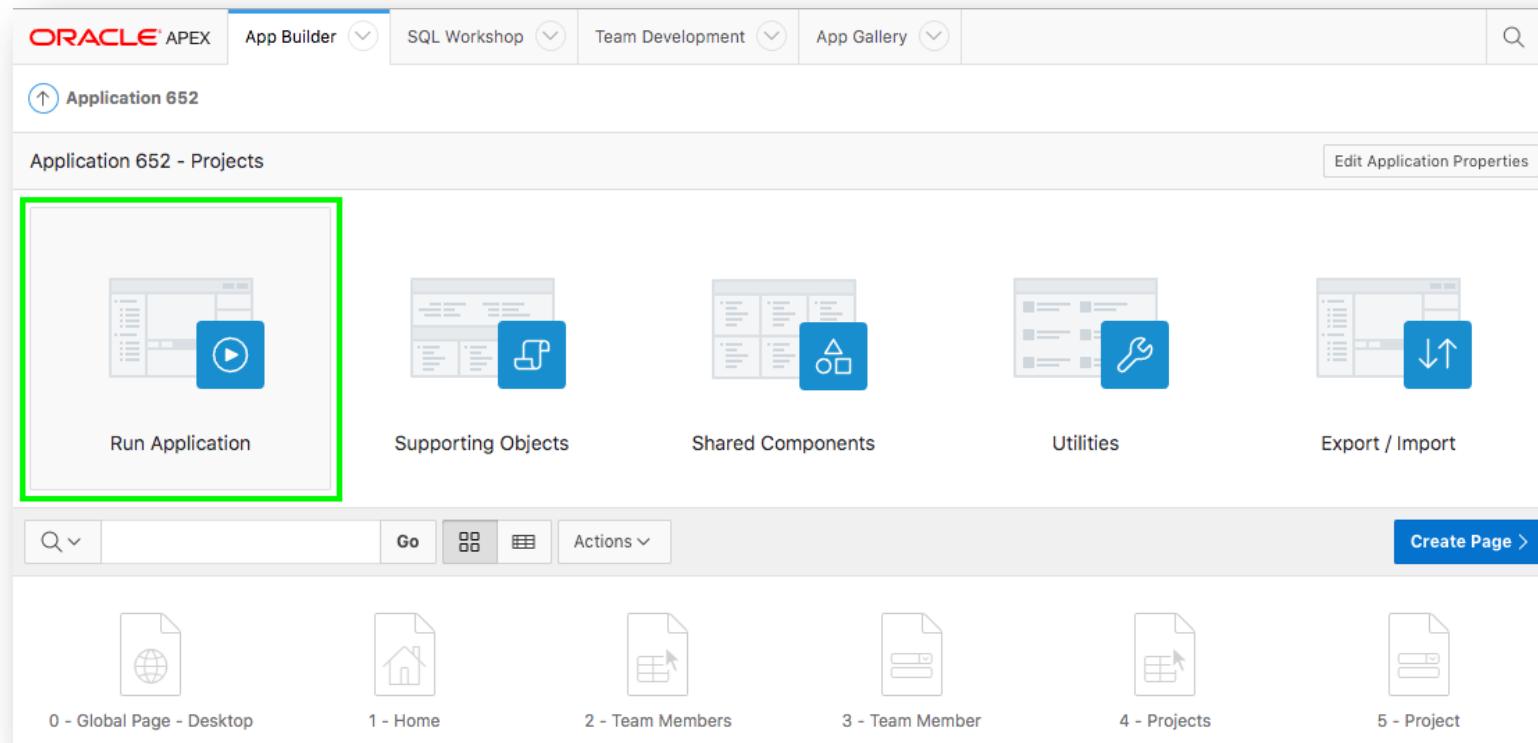
Step 3.4 – Create Application

- Click Create Application



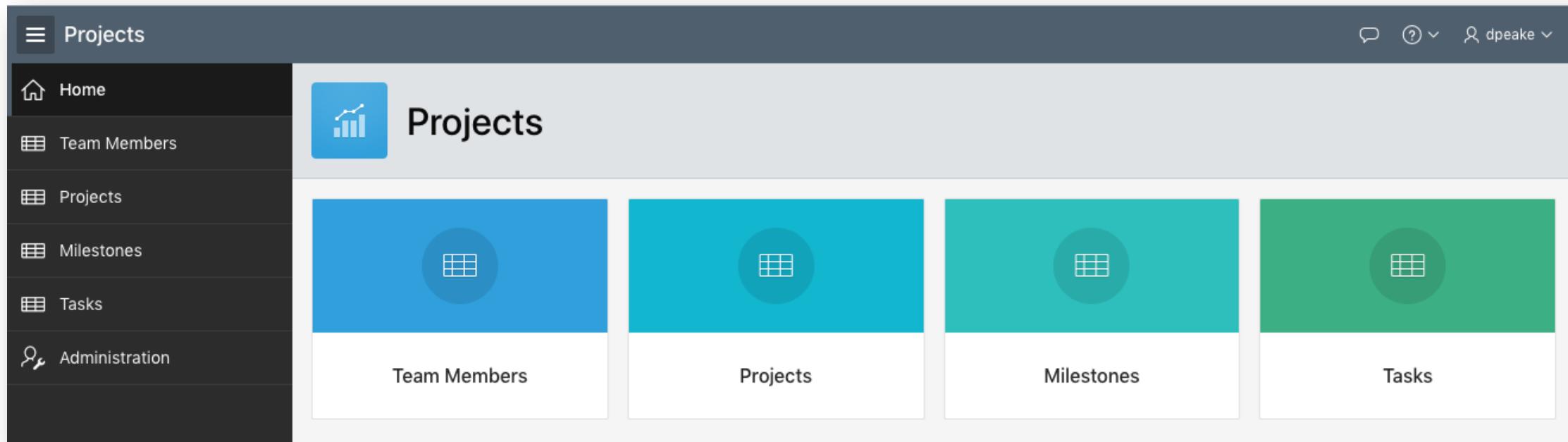
Step 3.5 – App in Page Designer

- Your new application will be displayed in Page Designer
- Click Run Application



Step 3.6 – Runtime App

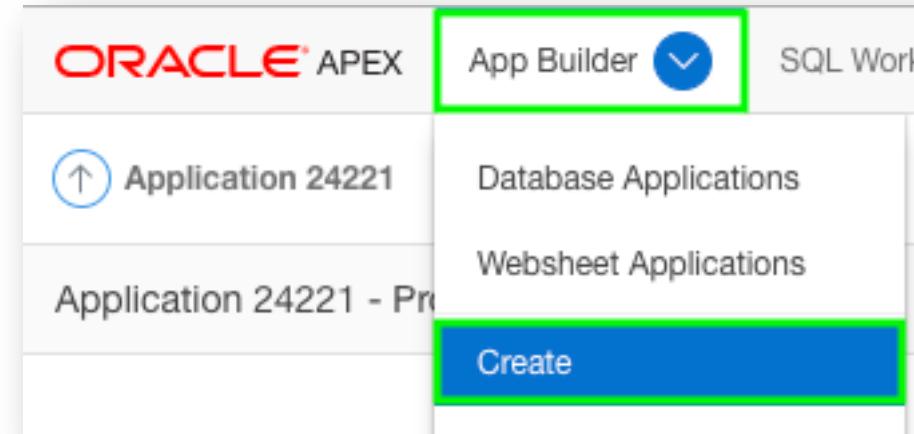
- Enter your credentials
- Play around with your new application



Improving the App Updating a Page

Step 4.1 – Restart the Create App Wizard

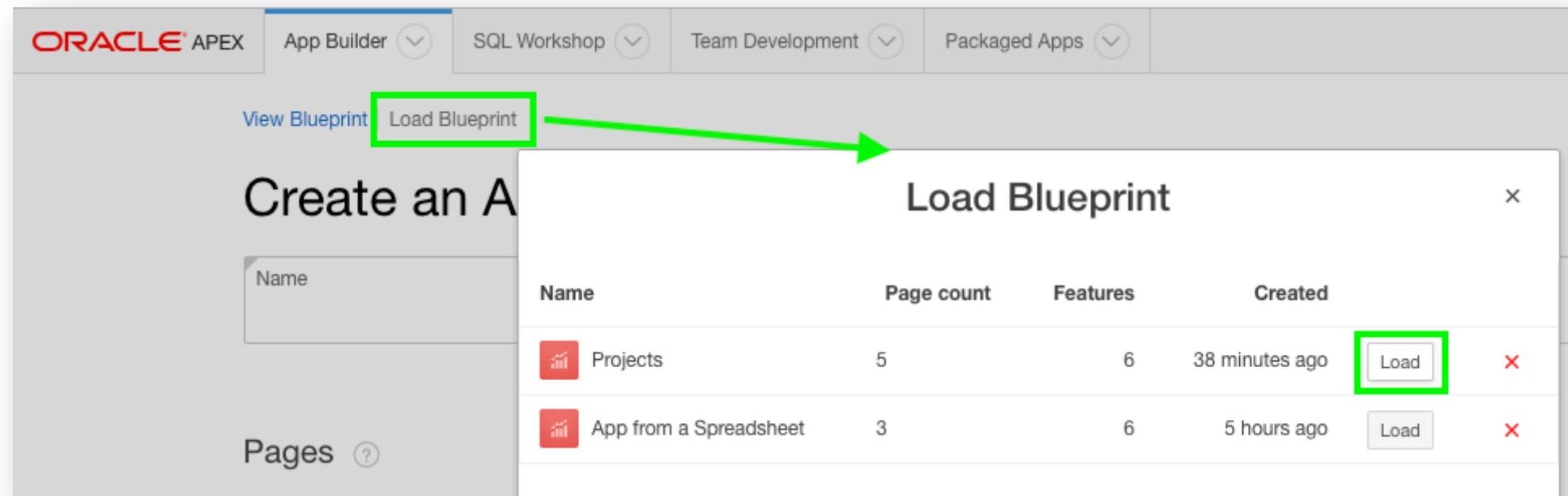
- From the development environment, click **App Builder**, and then select **Create**



- Click **New Application**

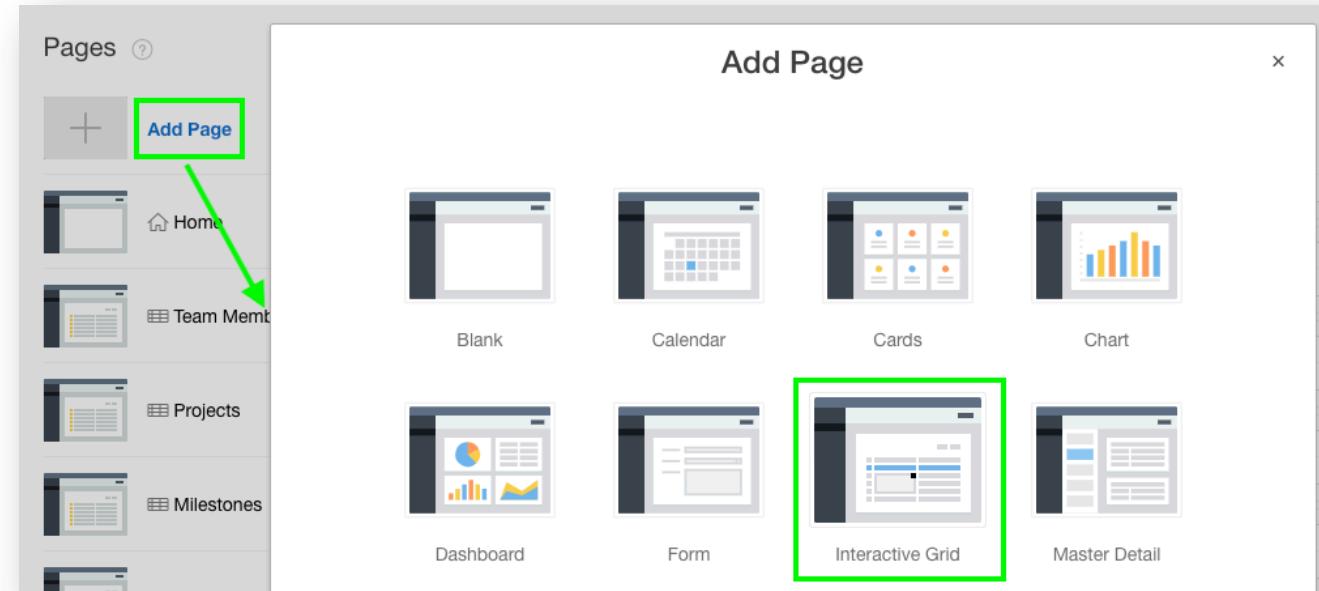
Step 4.2 – Load Blueprint

- In the Create App Wizard, click **Load Blueprint**
- For Projects, click **Load**

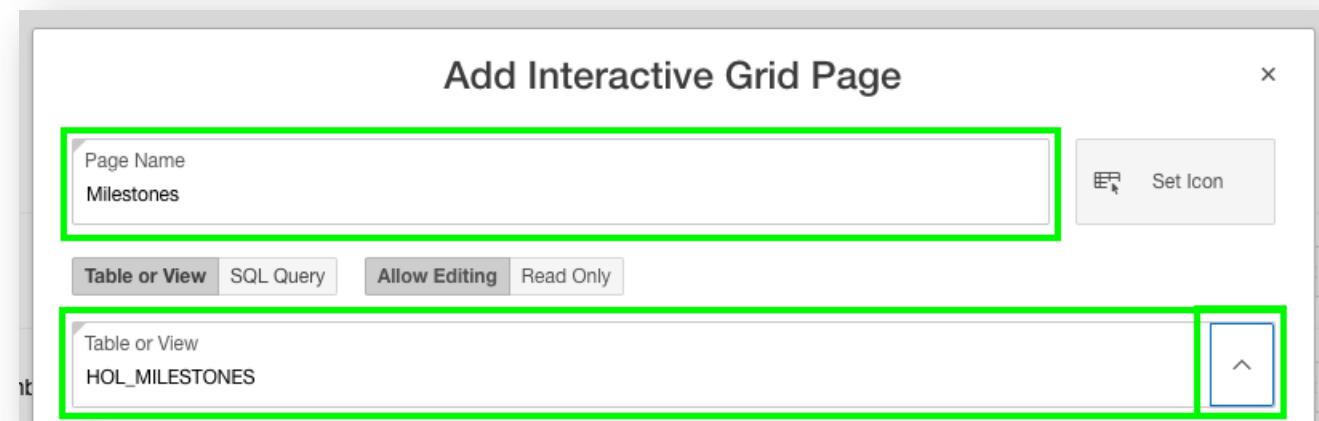


Step 4.3 – Add a Page

- Click Add Page
- Click Interactive Grid

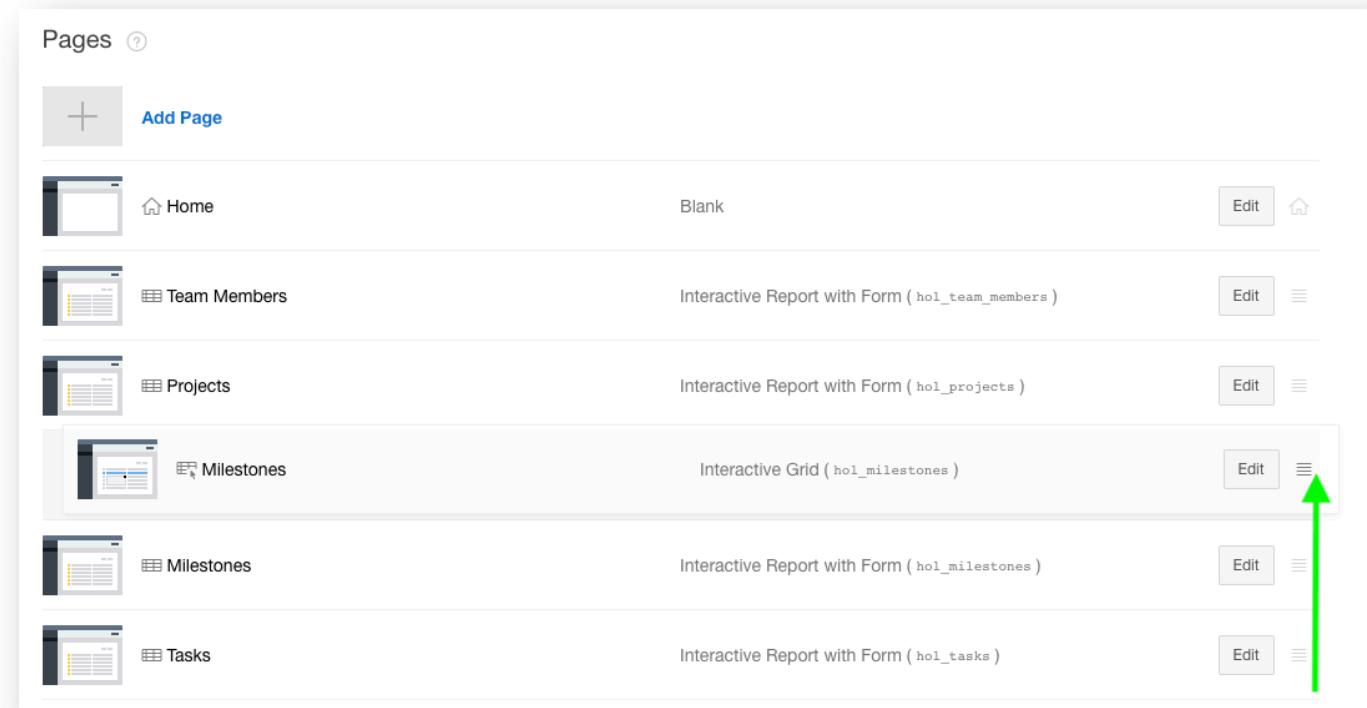


- For Page Name, enter **Milestones**
- For Table or View, select **HOL_MILESTONES**
- Click Add Page



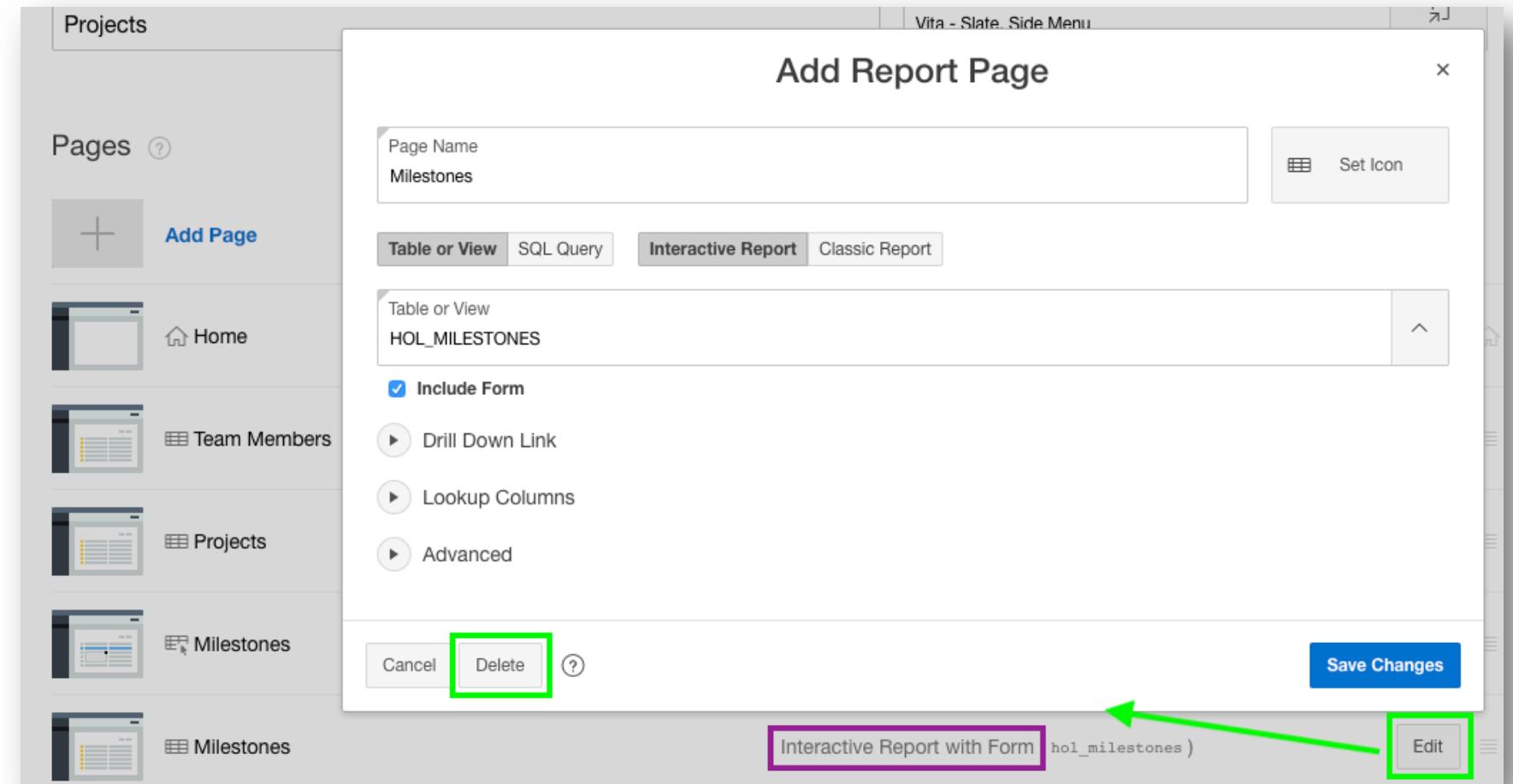
Step 4.4 – Reorder a Page

- Click and hold the mouse when hovering over the hamburger for the **Milestones – Interactive Grid** page
- Move it up until the page is under Projects
- Release the mouse



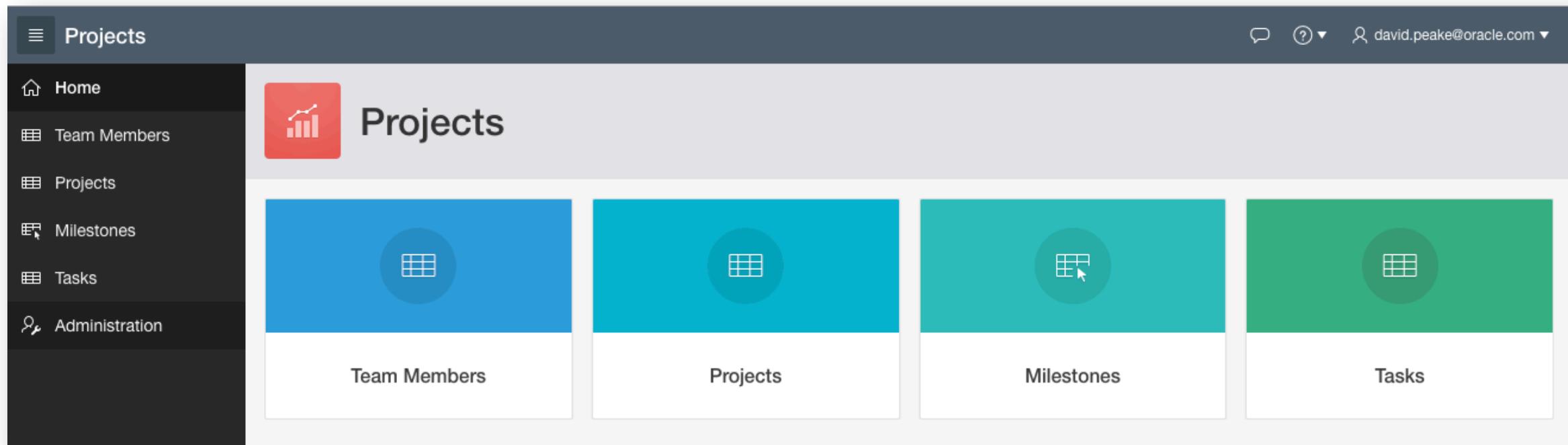
Step 4.5 – Delete a Page

- For Milestones – Interactive Report with Form page, click Edit
- Click Delete



Step 4.6 – Create App and Run

- Click Create Application
- In Page Designer, click Run Application



Step 4.7 – Navigate to Milestones

- In the runtime environment, click **Milestones**

The screenshot shows the Oracle ADF runtime environment. The left sidebar has a dark theme with the following menu items: Home, Team Members, Projects, **Milestones** (which is highlighted with a green border), Tasks, and Administration. The main content area is titled "Milestones" and contains a table with the following data:

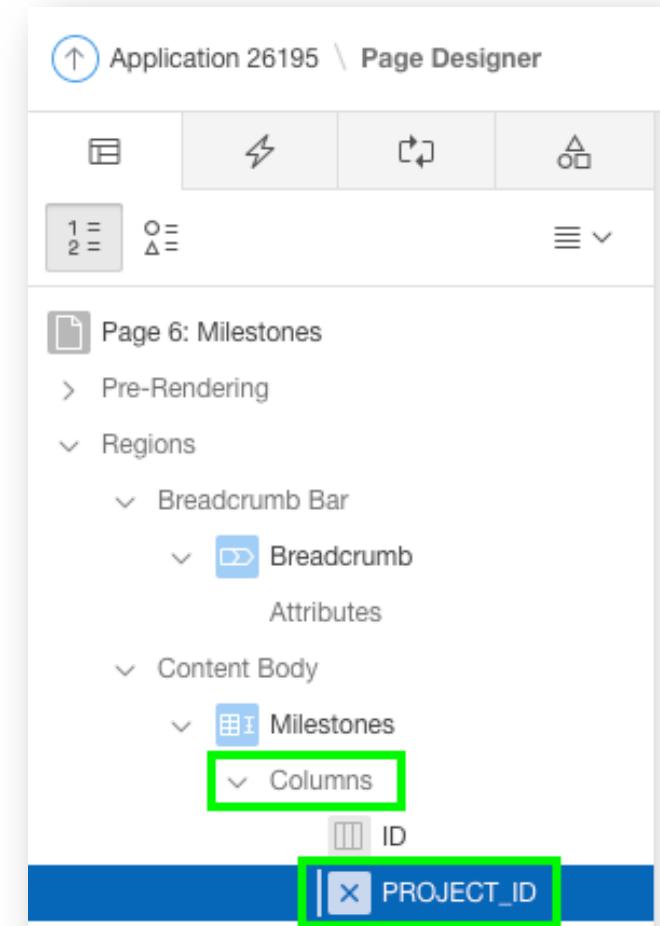
	Project Id	Row Version	Name	Due Date	Description	Created	Created By	Updated	Upda
<input checked="" type="checkbox"/>	8	1	Energy Efficiency	09-NOV-2017	Mi venenatis ne...	09-JUL-2018	DAVID.PEAKE@...	09-JUL-2018	DAVID
<input type="checkbox"/>	16	1	Transfer To Man...	09-AUG-2017	Nec. Donec con...	09-JUL-2018	DAVID.PEAKE@...	09-JUL-2018	DAVID

- In the Developer Toolbar, click **Edit Page 6**

The screenshot shows the Oracle ADF developer toolbar. The buttons from left to right are: Home, Application 24221, **Edit Page 6** (which is highlighted with a green border), Session, View Debug, Debug, Page Info, Quick Edit, Theme Roller, and a gear icon. Below the toolbar, there is a status bar showing "Total 30".

Step 4.8 – Update Project ID Column

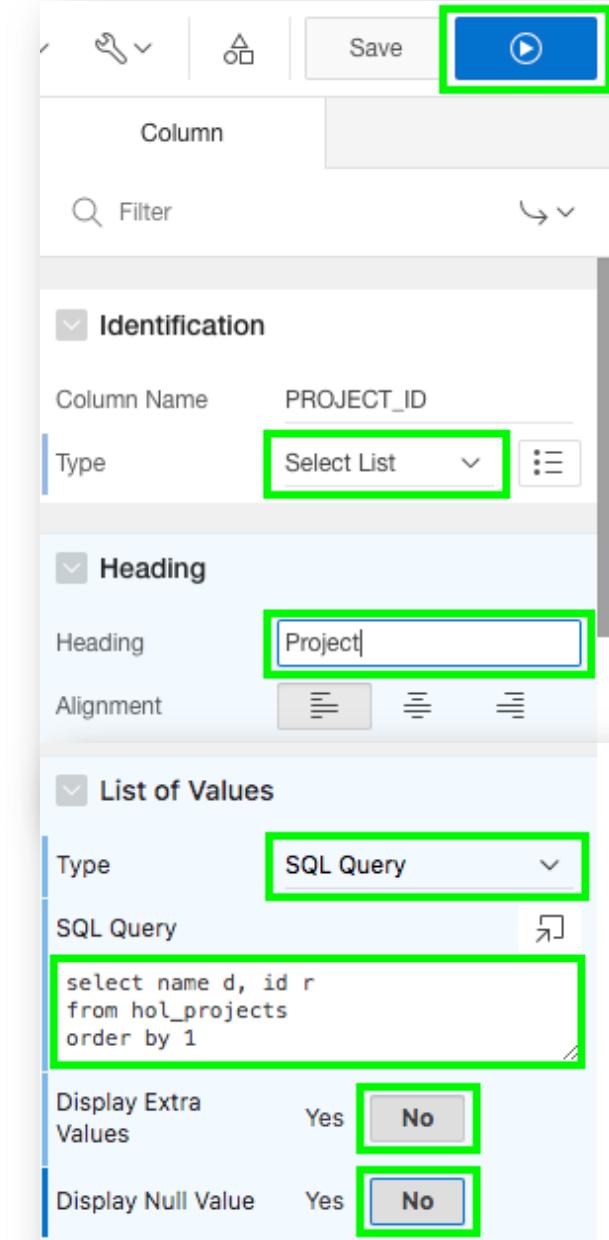
- In Page Designer, under Milestones, click Columns
- Click PROJECT_ID



Step 4.8b – Update Project ID Column

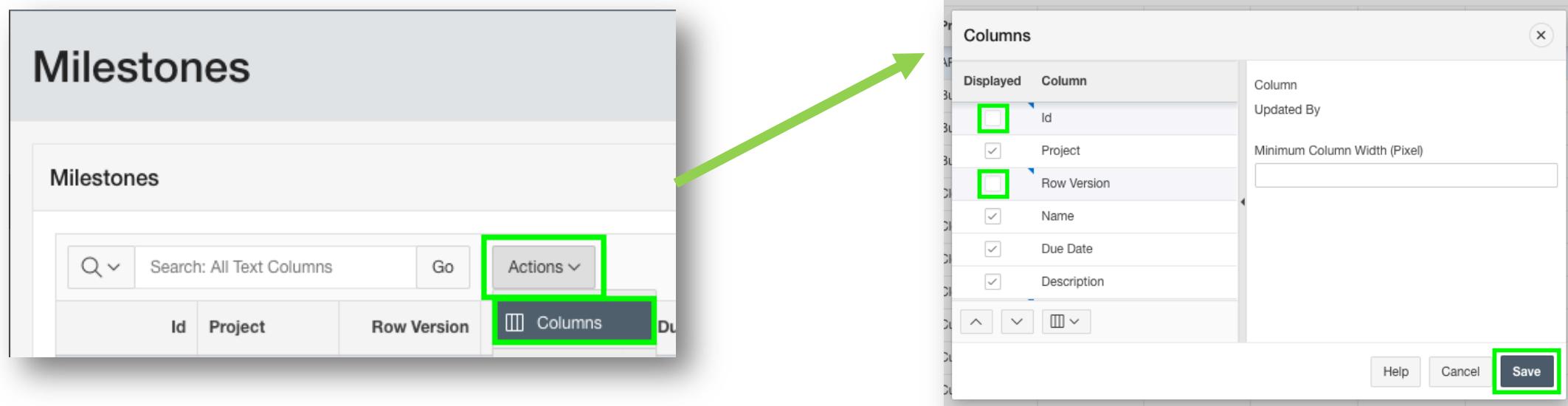
- In the Property Editor, update the following:
 - Identification: Type – select **Select List**
 - Heading: Heading – enter **Project**
 - List of Values: Type – select **SQL Query**
 - List of Values – SQL Query enter

```
select name d, id r
from hol_projects
order by 1
```
 - Display Extra Values – click **No**
 - Display Null Value – click **No**
- Save and Run the App



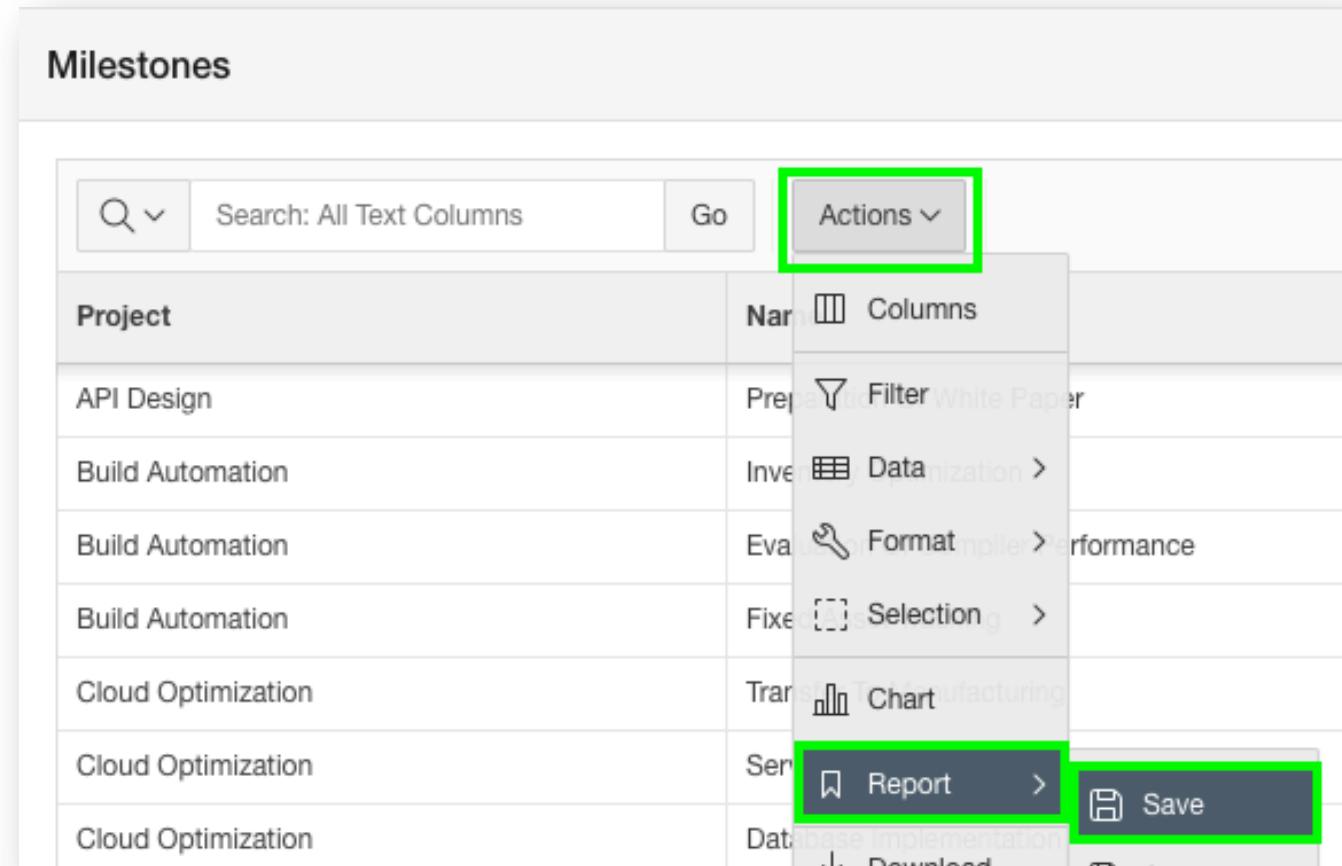
Step 4.9 – Hide Columns

- In the runtime environment, click **Actions**, select **Columns**
- Uncheck **Displayed** for **Id**, **Row Version**, **Created**, **Created By**, **Updated**, and **Updated By**
- Click **Save**



Step 4.10 – Save the Report

- In the runtime environment, click **Actions**, select **Report**, select **Save**



Learn More Useful Links

Useful Links

- APEX on Autonomous
- APEX Collateral
- Tutorials
- Community
- External Site + Slack

<https://apex.oracle.com/autonomous>

<https://apex.oracle.com>

<https://apex.oracle.com/en/learn/tutorials>

<https://apex.oracle.com/community>

<http://apex.world>

ORACLE®