Working with the Common Data Service

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\07_CommonDataService\Lab

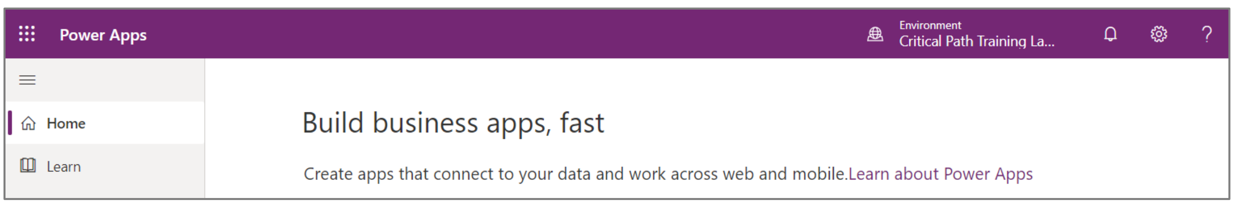
Lab Overview: In this lab, you will begin by using the Power Apps admin center to create a new Power Apps environment which is provisioned with a new CDS for Apps database. After creating the CDS for Apps database, you will inspect the set of standard entities and then you will create a canvas app to add, view and update contacts within the CDS. After that, you will use the new Power Apps preview support to create a model-driven app which is also designed to add, view and update contact data within the CDS database.

Lab Prerequisite: This lab assumes you completed lab 1 where you created a Power Apps per user plan trial subscription and assigned a Power Apps per user plan license to the primary Office 365 user account you have been using. If you have not already configured your Office 365 account with a Power Apps per user plan license, you complete all of exercise 2 of lab 1 before you begin working on this lab.

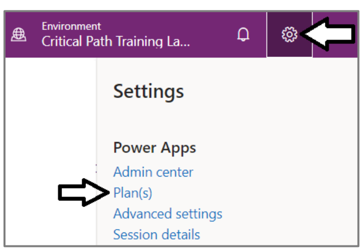
Exercise 1: Create a New Power Apps Environment with a CDS Database

In this exercise, you will create a new Power Apps environment with a new CDS database. After that, you will begin the provisioning process to create a Power Apps portal.

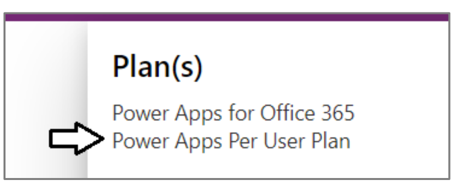
1. Make sure you have a **Power Apps per user** license.
   a) In the browser, navigate the Power Apps Maker portal at https://make.Power Apps.com.



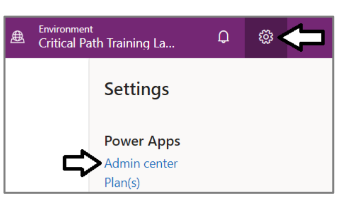
   b) Drop down the menu with the gear icon in the upper right of the page and select the **Plan(s)** command.



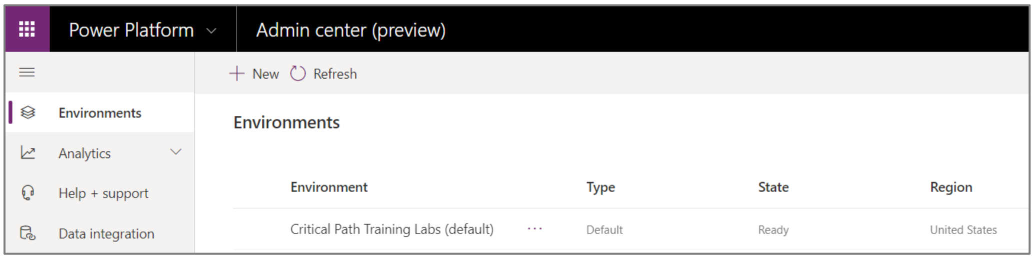
   c) You should be able to verify that you have a Power Apps Per User Plan.



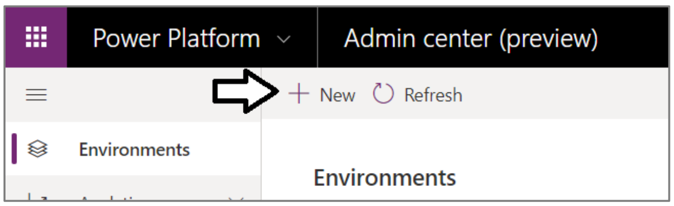
2. Create a new Power Apps environment with a Common data Service database.
   a) Drop down the menu with the gear icon in the upper right of the page and select the **Admin center** command.



b) In the **Power Platform Admin center**, click the **Environments** link in the left navigation.



c) Click the **+ New** button to create a new environment.

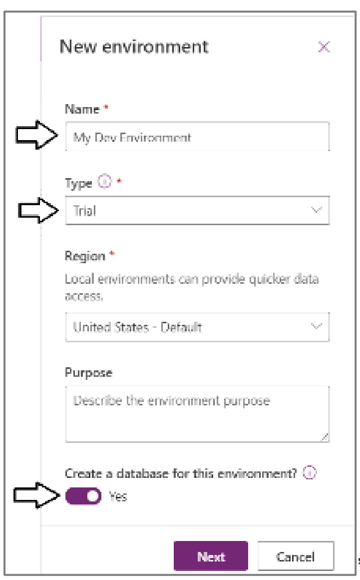


d) In the **New environment** pane on the right, enter a Name of **My Dev Environment**.

e) Leave the environment **Type** setting with its default value of **Trial**.

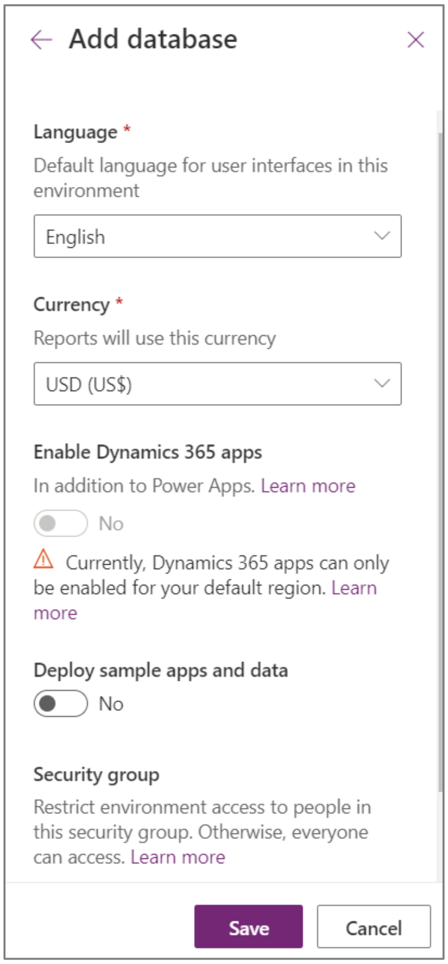
f) Make sure you enable the option **Create a database for this environment**.

g) Once you have filled out the **New environment** pane, click **Next**.

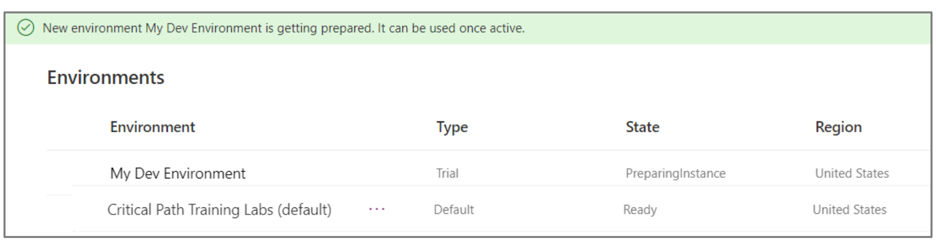


As of January of 2020, the Power Platform only allows users with a trial licenses to create trial environments. You will need to paid license to create a production environment or to promote a trial environment to a production environment

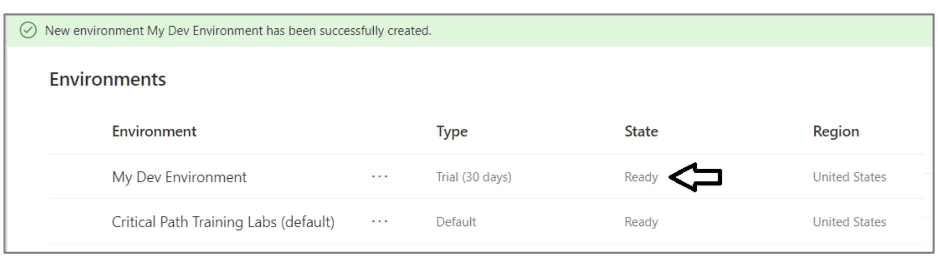
h) Leave all the default settings on the **Add database** pane as shown in the following screenshot and click **Save**.



i) You should now see the new environment in the tenant **Environments** list with a **State** of **PreparingInstance**.

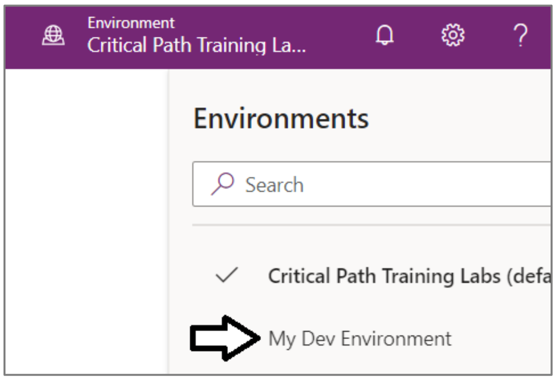


j) Refresh the **Environments** list every 30 seconds or so until you see that he **State** has turned to **Ready**.



3. Move the Power Apps maker portal over to work with your new Power Apps environment.
   a) Return to the Power Apps Maker portal at https://make.Power Apps.com.
   b) Refresh the page at https://make.Power Apps.com by pressing **{F5}**.
   c) Click the **Environment** menu at the top right of the page to drop down the environment selector menu.

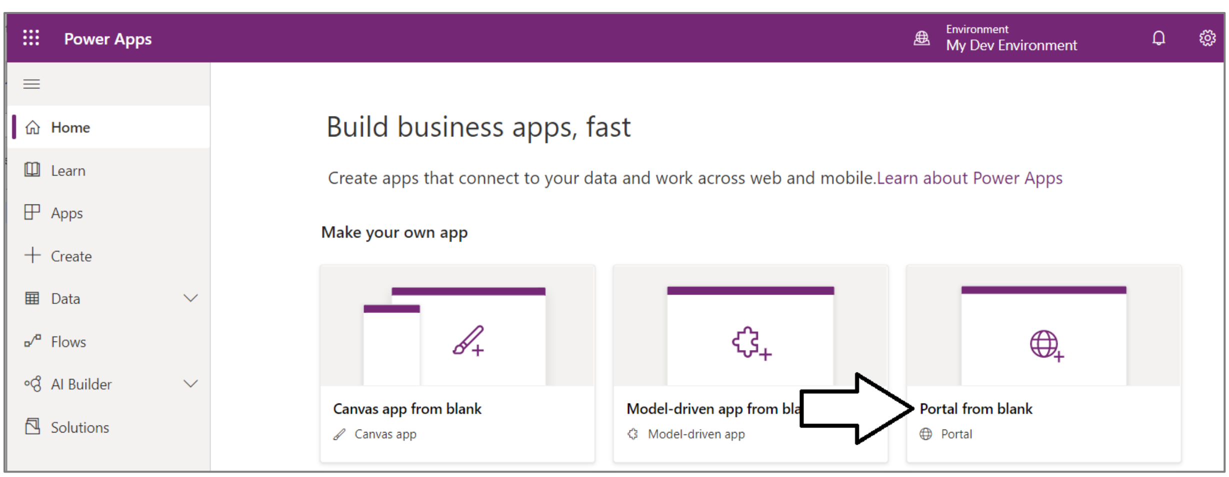
   d) Switch over to the new environment you just created named **My Dev Environment**.



   e) You should be able to confirm that the current environment has been switched over to **My Dev Environment**.

It is not yet time for you to begin work with a Power Apps portal. However, it takes a long time to create a Power Apps portal. Therefore, you are going to begin the process to create a Power Apps portal now so it will be ready later on in a future lab exercise.

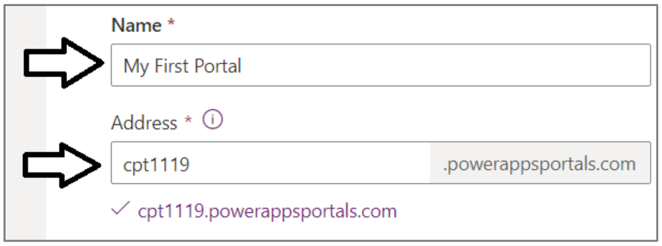
4. Create a Power Apps portal in your new Power Apps environment.
   a) Click on the **Portal from blank (preview)** tile to create a new Power Apps portal.



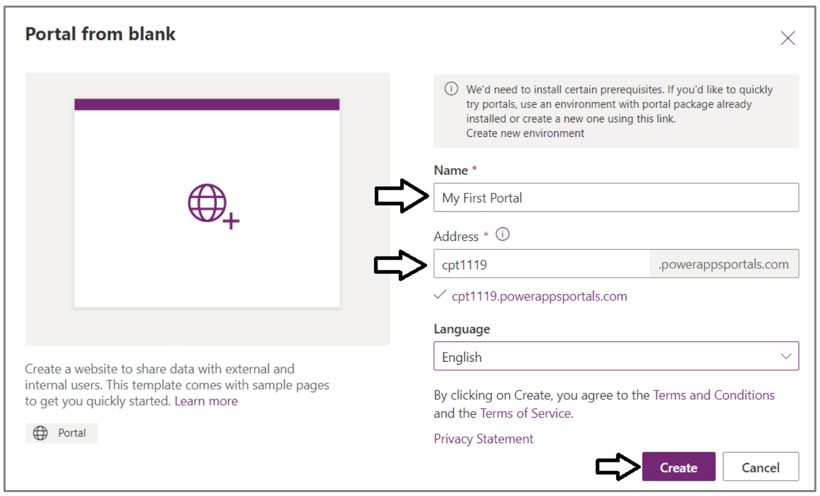
You can only create one Power Apps portal per environment. If you want to create multiple portals, you will have to create a separate Power Apps environment for each one.

b) In the **Portal from blank (preview)** dialog, enter a portal tile such as **My Portal Lab**.

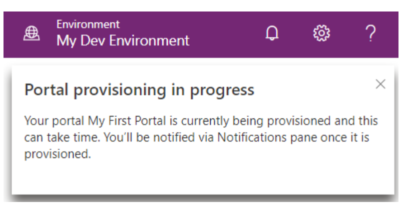
c) Enter a value for the portal **Address**. You might need to try several times if you pick an address that is already taken.



d) Once you have entered a **Name** and **Address** for your portal, click **Create** to begin the portal provisioning process.



e) You should now see a notification indicating that portal provisioning is now in process.

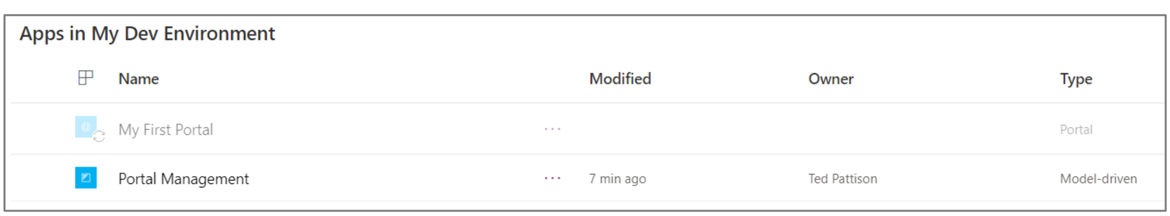


5. Inspect the apps in **My Dev Environment**.

a) Click the **Apps** link in the left navigation to see the list of apps for **My Dev Environment**.

b) You should see the portal app in a greyed-out site which indicates that the portal is being provisioned.

c) Provisioning a Power Apps portal will also eventually create another model-driven app named **Portal Management**.



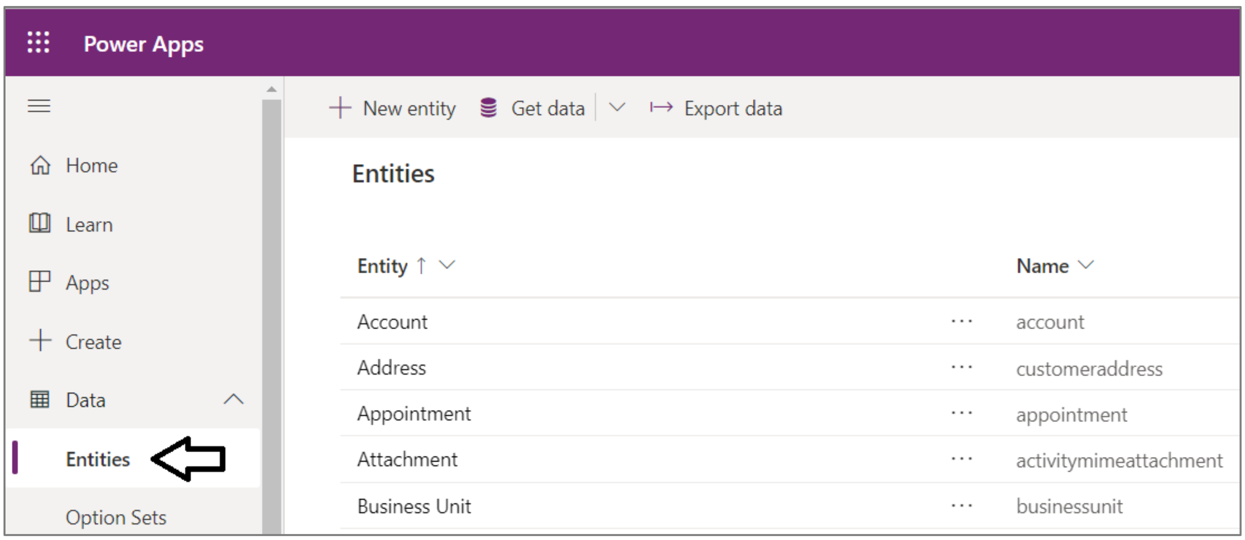
It will take quite a bit of time before your portal is ready to use and edit. The entire provisioning process for a new Power Apps portal can take from 30 to 60 minutes or even longer. You will now create a canvas app and model-driven app that are not related to this portal. However, you have started the Power Apps portal provisioning process so the portal will be later when it's time to work on it.

Exercise 2: Inspect Standard Entities in the Common Data Service

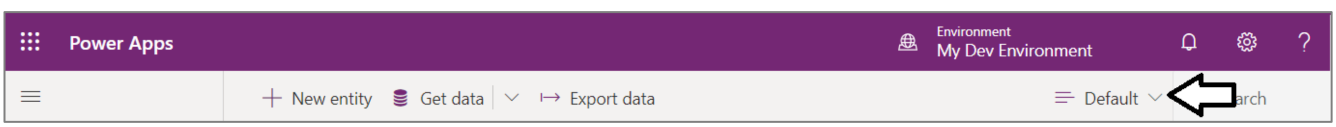
In this exercise, you will inspect some of the standard entities that are built into the CDS database.

1. Inspect the standard entities in the Common data Service.
   a) Navigate to the Power Apps home page at https://make.Power Apps.com.
   b) Make sure the current environment is the environment named **My Dev Environment** that you created in exercise 1.

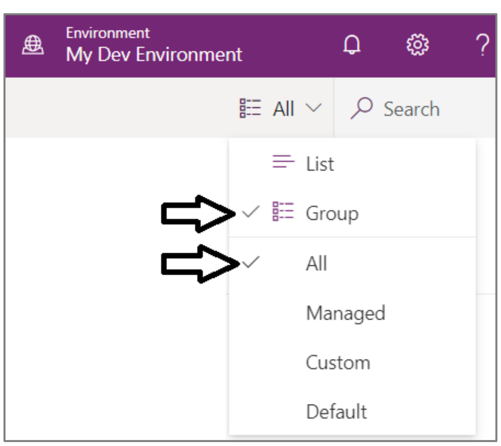
   c) Click on the **Entities** link in the **Data** section of the left navigation and examine the entities that are displayed.



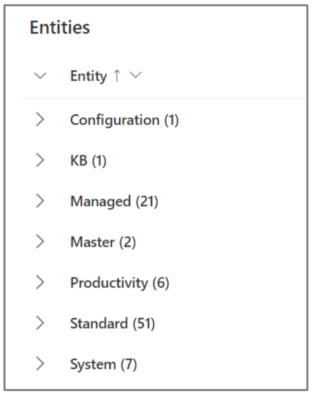
   d) Locate the view menu for the **Entities** page in the upper, right corner which should initially be set to **Default**.



   e) Drop down the view menu and select **Group** and **All** to display all entities separated into groups.



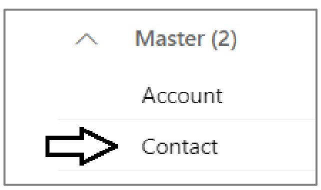
f) You should now see all the standard entities are organized in groups.



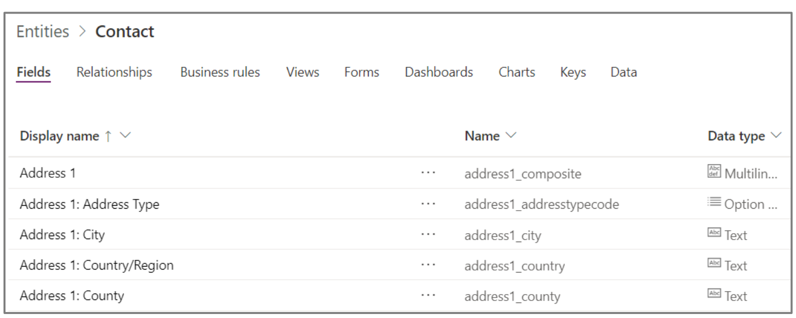
g) Expand the **Master** group and you should see this group contains two important entities named **Account** and **Contact**.

2. Take a closer look at the **Contact** entity.

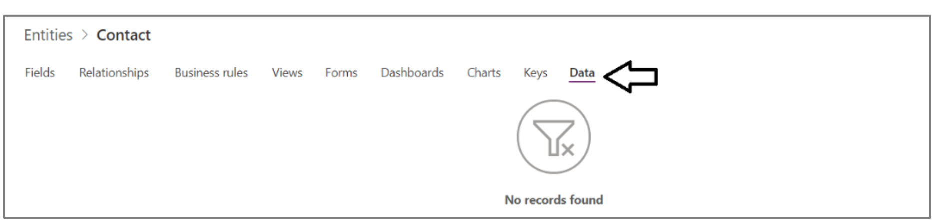
a) Click on the link for the **Contact** entity.



b) Examine the Fields list to see what fields are included with the **Contact** entity.



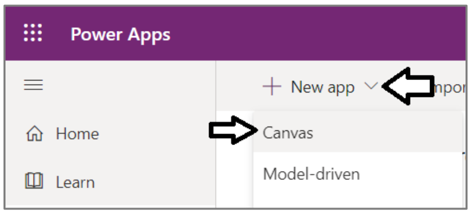
c) Click the **Data** tab to verify that the **Contact** entity currently contains no data.



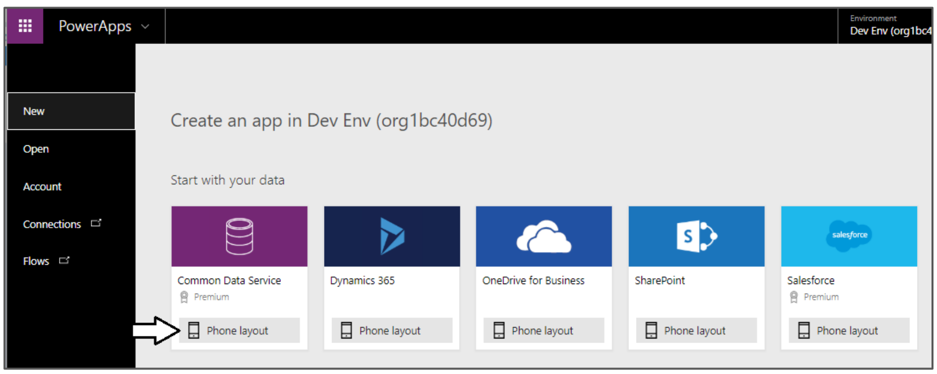
Exercise 3: Create a Canvas App to Manage Contact Entity Data

In this exercise, you will create a canvas app to manage **Contact** entity data.

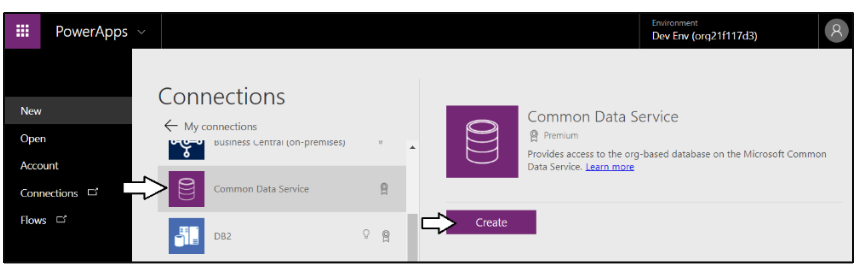
1. Create the new Canvas app using Power Apps Studio.
   a) In the Power Apps portal at https://web.Power Apps.com, make sure you are running in the **Dev Env** environment.
   b) Click the **Apps** link in the left navigation. The Apps list should be empty at this point.
   c) Click the **+ Create an app > Canvas** menu command to begin the process of creating a new canvas app.



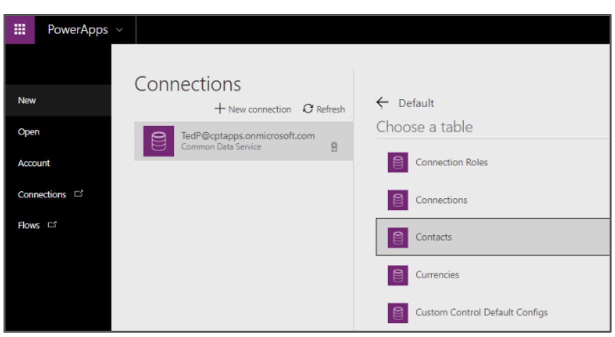
   d) On the **Create an app page**, locate the **Common Data Service** tile and click the **Phone layout** button inside it.



   e) When prompted to create a new Common Data Service connection, click the **Create** button to continue.

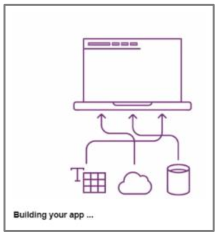


   f) In the **Choose a table** section on the **Connections** page, select the **Contacts** table.

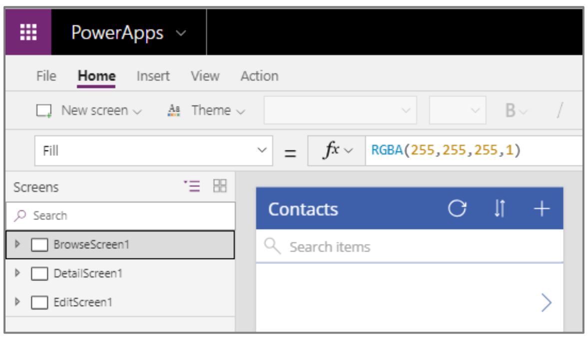


g) Once you have selected the **Contacts** table, click the **Connect** button.

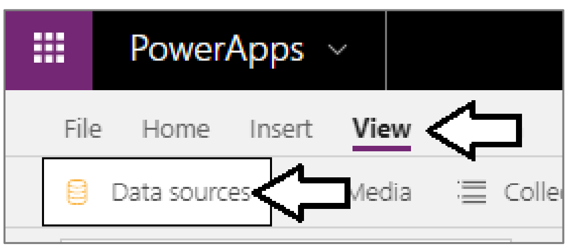
h) At this point, Power Apps will take 20-30 seconds to create the new Power Apps project.



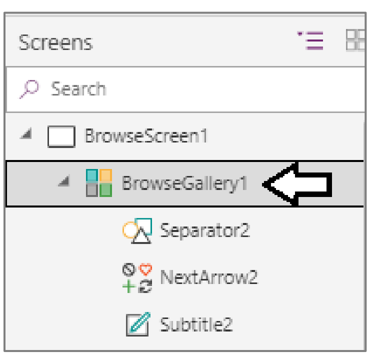
i) When Power Apps has finished creating the new canvas app project, you should see the app project has been created with a browse screen, a view detail screen and an edit screen as shown in the following screenshot.



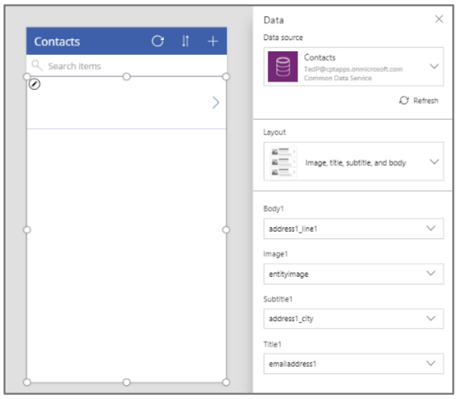
j) Select **BrowseScreen1** in the left navigation. Click **View** and then click **Data sources** in the ribbon to display the **Data** pane.



k) With the **Data** pane showing, select the **BrowseGallery1** control in the left navigation.

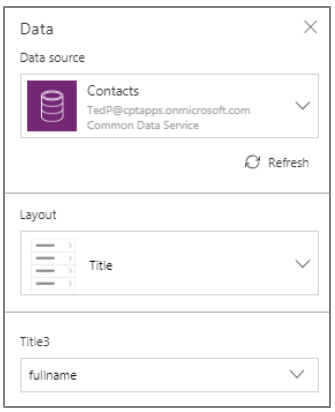


l) The Data pane should now display the **Layout** and fields used by the gallery control.

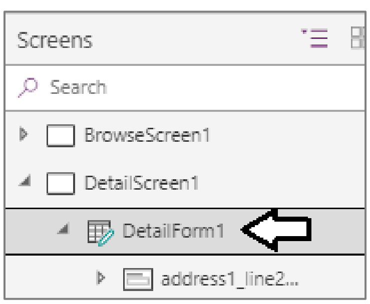


m) Update the Layout for the gallery to **Title**.

n) Configure the **Title3** gallery control to bind to the CDS field named **fullname** as shown in the following screenshot.



o) With the **Data** pane showing, use the left navigation to select **DetailForm1** in **DetailScreeen1**



p) The **Data** pane should now display a **Fields** list showed the fields included on **DetailForm1**.

q) Reorganize the fields in the Fields list to display the following fields in this particular order.

i) Contact

ii) Full Name

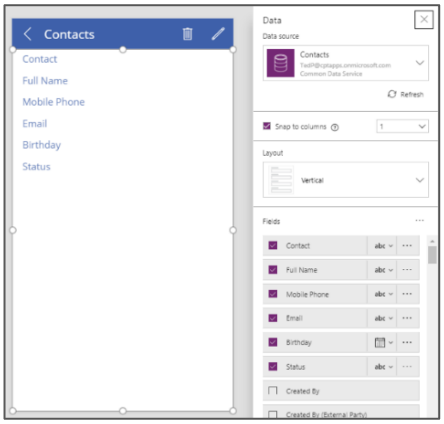
iii) Mobile Phone

iv) Email

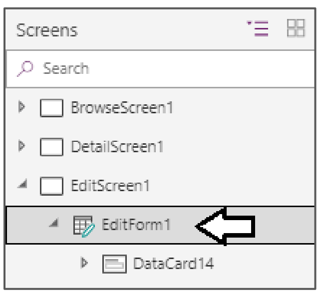
v) Birthday

vi) Status

r) The **Fields** list for **DetailForm1** should match the **Fields** list shown in the following screenshot.



s) With the **Data** pane showing, use the left navigation to select **EditForm1** in **EditScreeen1**



t) The **Data** pane should now display a **Fields** list showed the fields included on **EditForm1**.

u) Reorganize the fields in the **Fields** list for **EditForm1** to include the following fields in this particular order.

i) Last Name

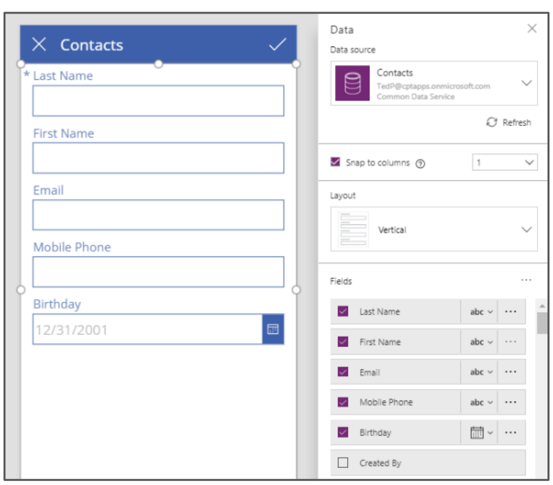
ii) First name

iii) Email

iv) Mobile Phone

v) Birthday

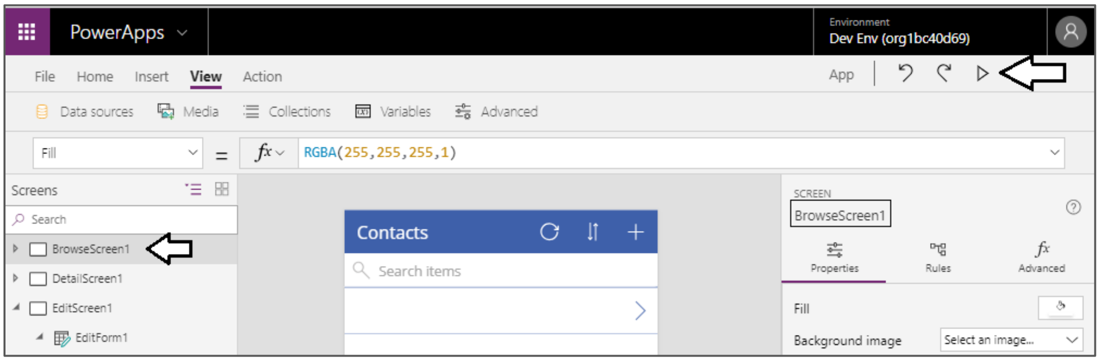
v) The **Fields** list for **EditForm1** should match the **Fields** list shown in the following screenshot.



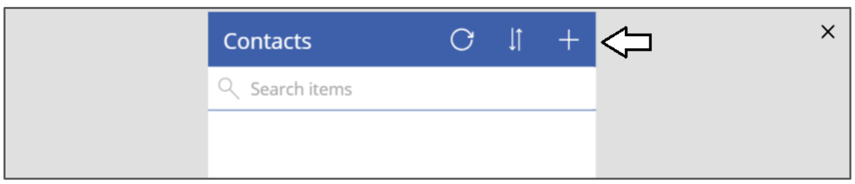
You have now completed building the new app and now it's time to test it out.

2. Test it out the application by running it and adding a new contact.

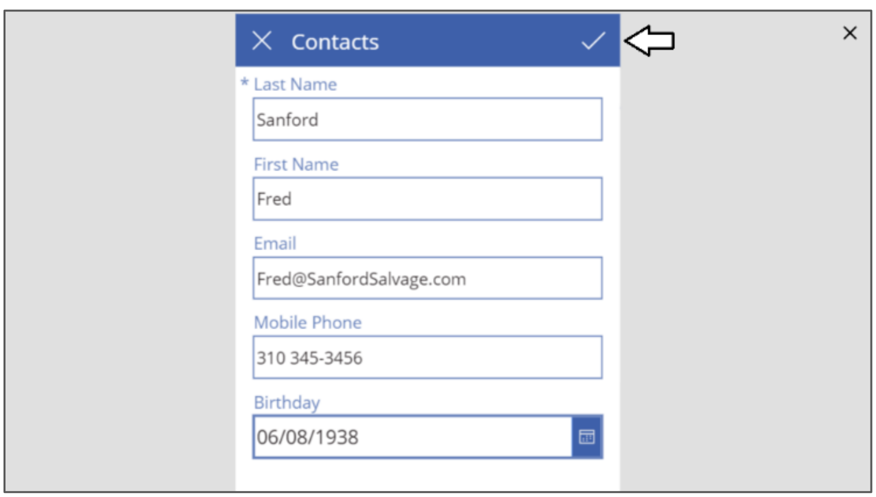
a) Select the startup screen named **BrowseScreen1** in the left navigation and then click the **Play** button to run the app.



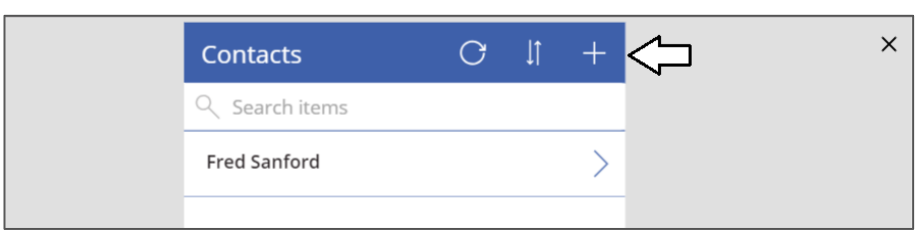
b) When the app starts and display the startup browse screen, click the **+** button to create a new contact.



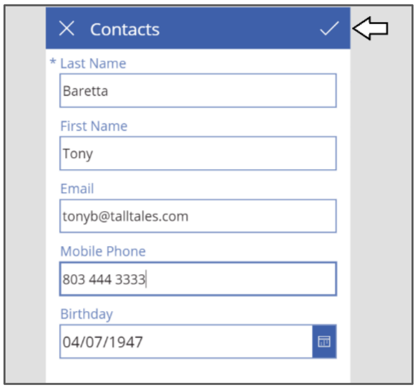
c) Enter sample data for a new contact like the data shown in the following screenshot and then click the Checkmark button to save your changes and create a new contact.



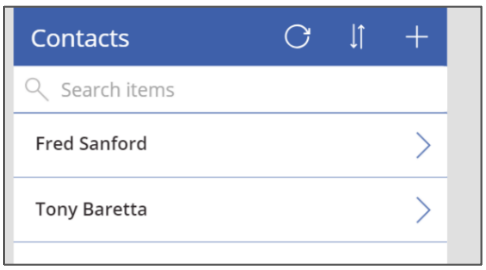
d) Once the new contact is created, the app should redirect you back to the browse screen which displays the new contact.



e) Follow the same steps to create and save a second contact in the CDS for Apps database.

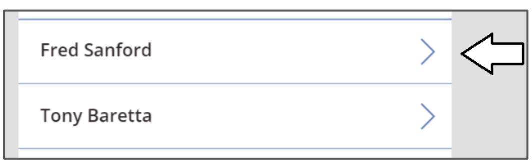


f) When you are done, you should now see two new contacts in the browse screen.

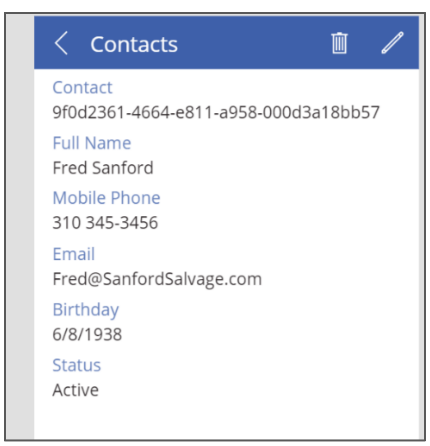


3. Test the details screen.

a) Click the **>** button for one of the new contacts to navigate to the app's detail screen.



b) On the detail screen, you should see the details for a specific contact.

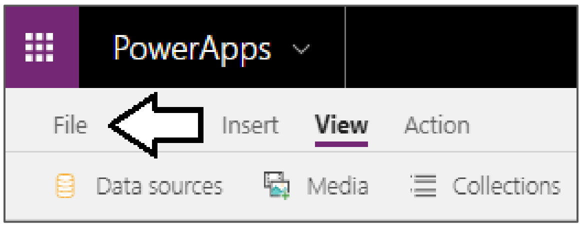


Note that the **Contact** field contains a GUID that serves as the ID (*i.e. primary key*) for the contact record. Generating a new GUID and assigning it to the **Contact** column is handled behind the scenes by the Common Data Service whenever a new contact is created.

c) Quit the running app by clicking the **x** button in the upper right corner of the page to return to your app in Power Apps Studio.

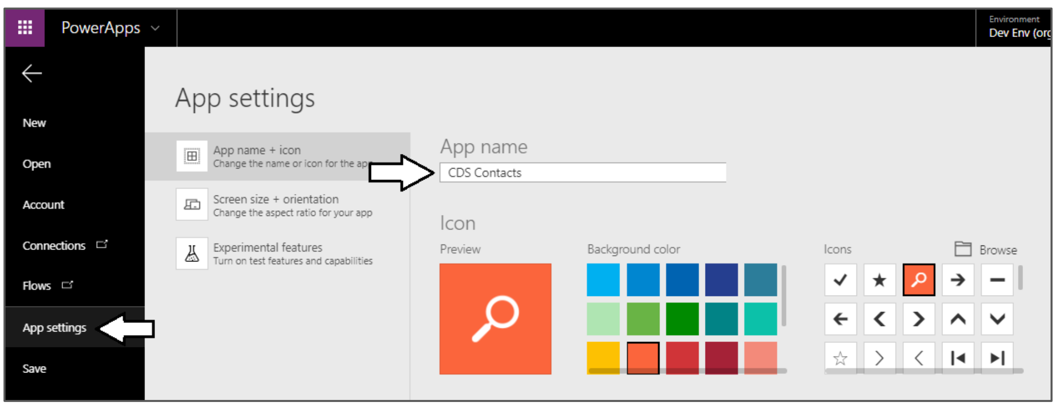
4. Save your work by saving the app in the cloud.

a) In Power Apps Studio, click the **File** menu.

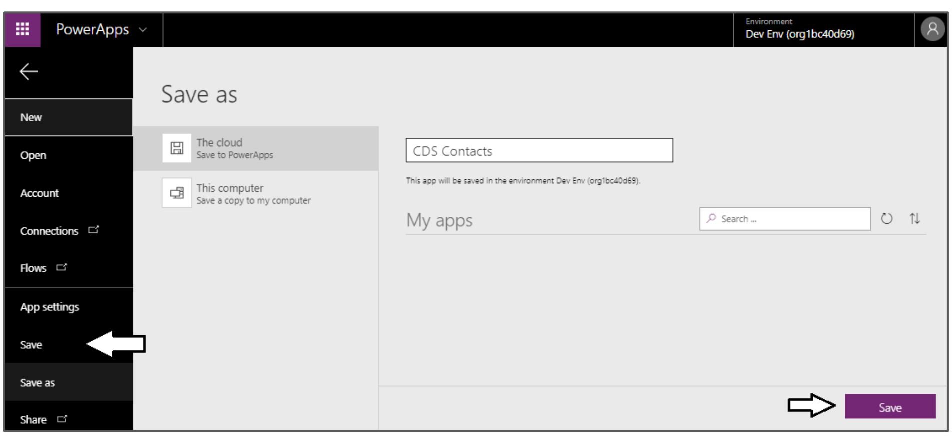


b) Click on the **App settings** link in the left navigation and enter an **App name** of **CDS Contacts**.

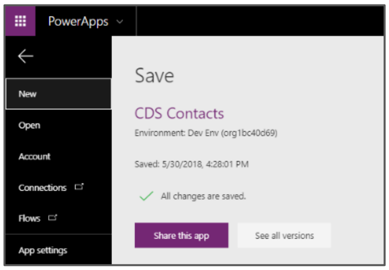
c) Select an **Icon** and a **Background color** of your liking.



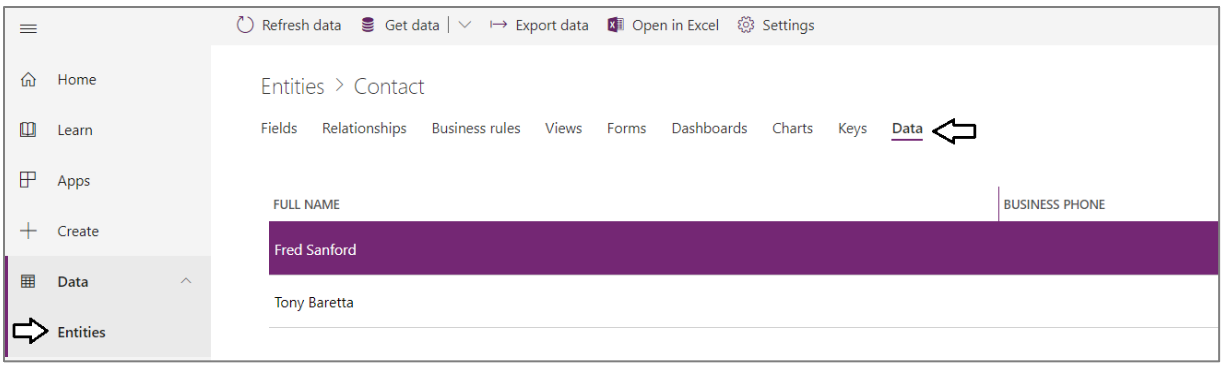
d) Click the **Save** link in the left navigation and then click the **Save** button in the bottom right corner of the **Save as** page.



e) The Save page should now display a message indicating that *All changes ae saved*.



5. Inspect the Data view of the **Contact** entity.
   a) Return to the Power Apps portal and click the Data > Entities
   b) Locate and click he **Contact** entity.
   c) Click the **Data** link at the first right to see your new contacts.

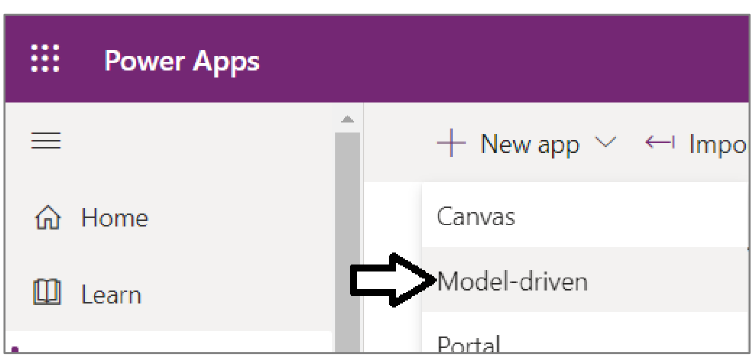


You have now created a classic-style app in Power Apps which is now known as a Canvas app. Next, you will create a second app using the new model-driven app support. While model-driven apps are still in preview, it will give you a good sense of the difference between canvas apps and model-driven apps.

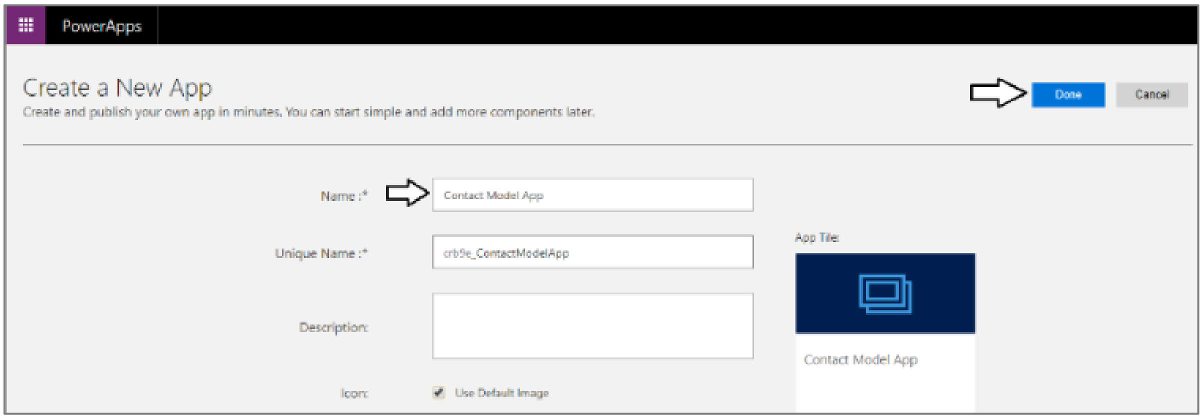
Exercise 4: Create a Model-driven App to Manage Contact Entity Data

In this exercise, you will create a new model-driven app using the new preview support in Power Apps Studio.

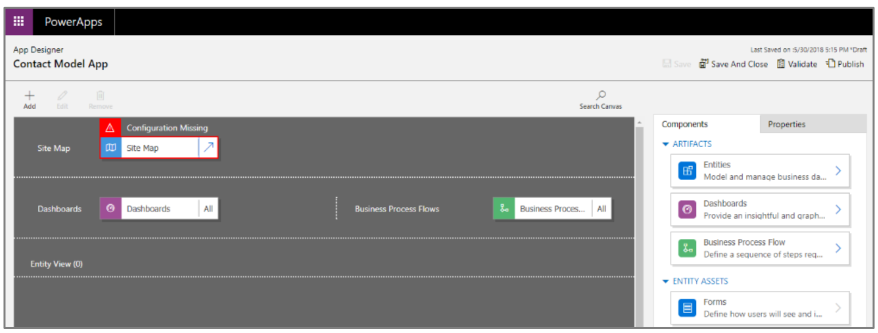
1. Create the new Canvas app using Power Apps Studio.
   a) In the Power Apps portal at https://web.Power Apps.com, make sure you are running in the **Dev Env** environment.
   b) Click the **Apps** link in the left navigation.
   c) Click the **+ Create an app > Model-driven** menu command to begin the process of creating a new model-driven app.



d) On the **Create a New App** page, enter a **Name** of **Contact Model App** and then click the **Done** button.



2. You should now see the new model-drive app in the new preview **App Designer** as shown in the following screenshot.

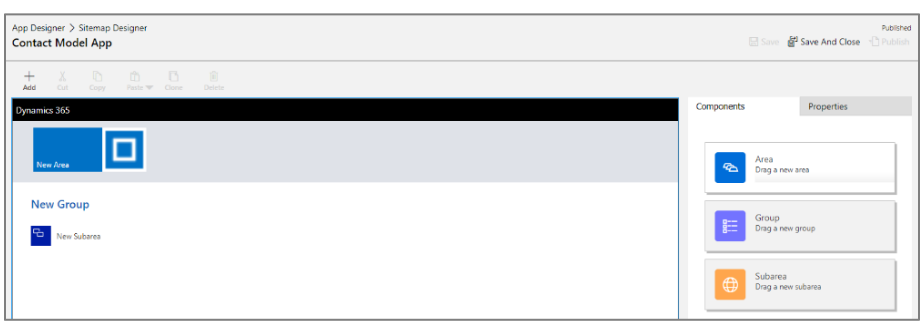


Note that the app is not ready to run because there is a red error indicator above the app's site map component. You must make some changes to the site map before you can run this model-driven app.

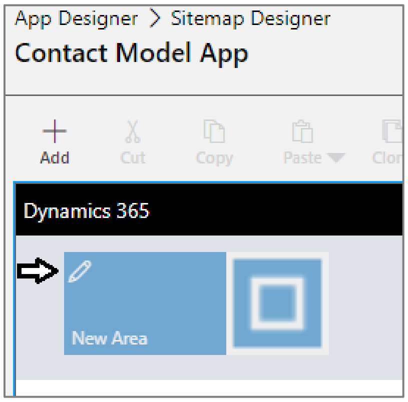
3. Configure the model-driven app's site map.

a) Click on the button with the arrow on the left side of the Site Map to open the site map in the Site Map Designer.

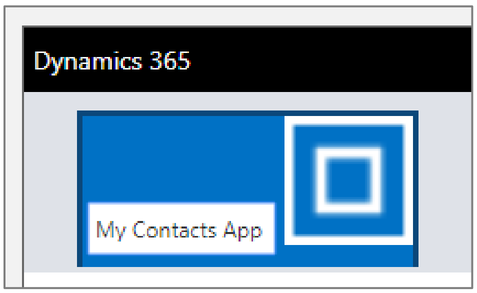
b) You should now see the site map for your app open in the **Site Map Designer**.



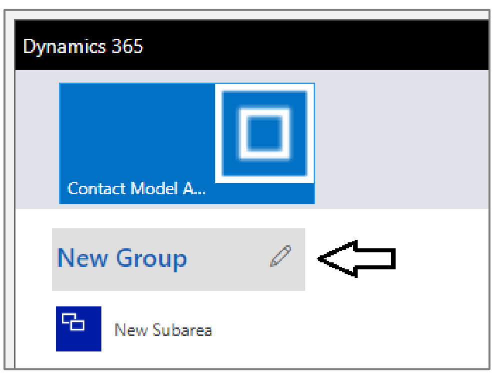
c) In the Site Map Designer, hover the mouse over **New Area** and click the pen icon to move the area into edit mode.



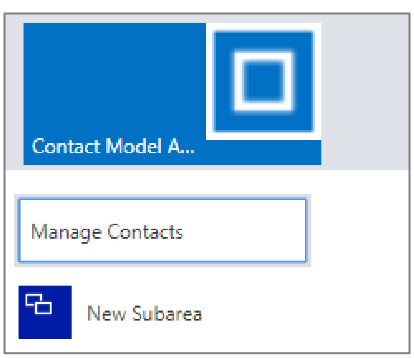
d) Change the text caption for the area from **New Area** to **My Contacts App**.



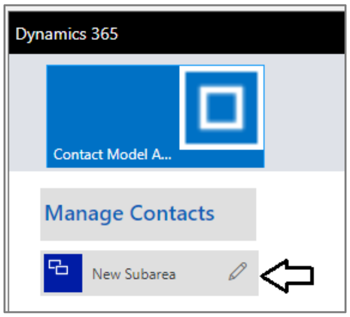
e) Hover the mouse over the group named **New Group** and click the pen icon to move the group into edit mode.



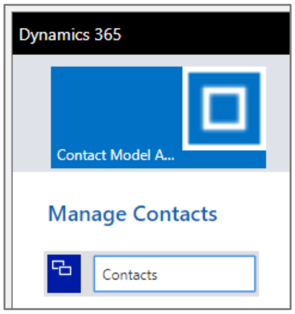
f) Change the text caption of the group from **New Group** to **Manage Contacts**.



g) Hover the mouse over the subarea named **New Subarea** and click the pen icon to move the subarea into edit mode.



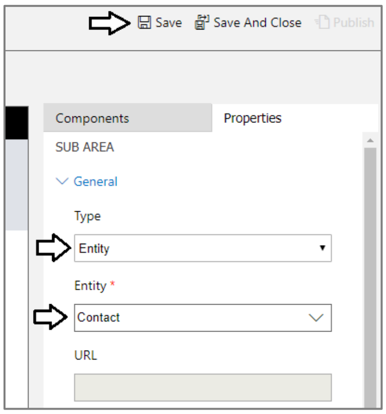
h) Change the text caption of the subarea from **New Subarea** to **Contacts**.



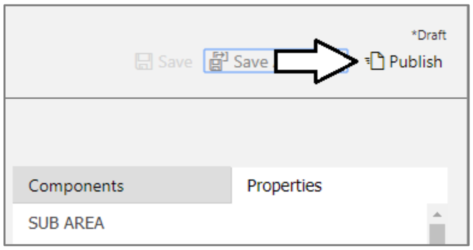
i) Select the **Contacts** subarea with the mouse and then inspect the **Properties** pane on the right side of the browser window. You will see a few red messages indicating that you must make some changes before the site map will work correctly.

j) In the **Properties** pane for the **Contacts** subarea, set the **Type** property to **Entity** and then set **Entity** property to **Contact**.

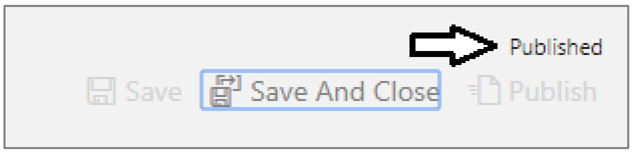
k) Click the **Save** button at the top of the **Properties** pane to save your changes to the app's site map.



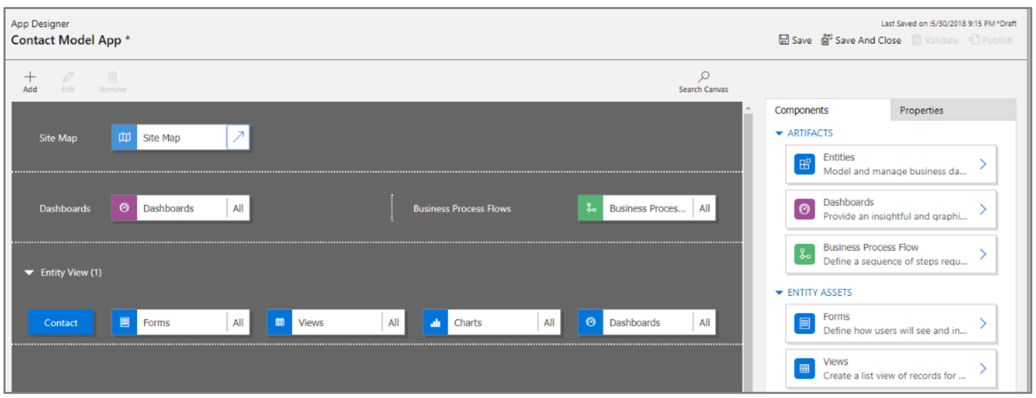
l) After you have saved the site map, click the **Publish** button to publish the site map.



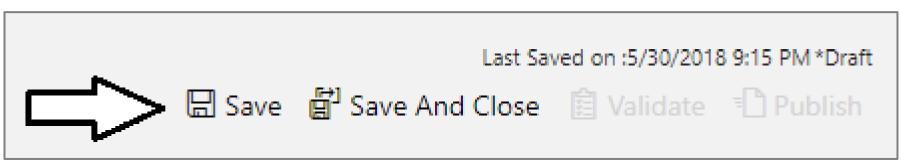
m) After publishing, you should see a message indicating the site map has been published. Click the **Save and Close** button to close the **Site Map Designer** and navigate back to the **App Designer**.



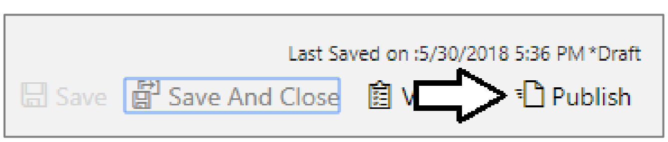
n) You should now see the App Designer and the red error indicators that were displayed in the site map earlier should be gone.



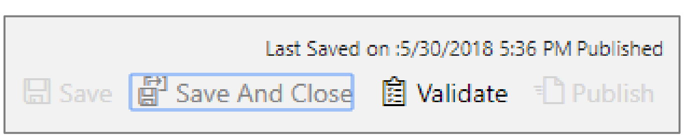
o) Save the new app by clicking the **Save** button.



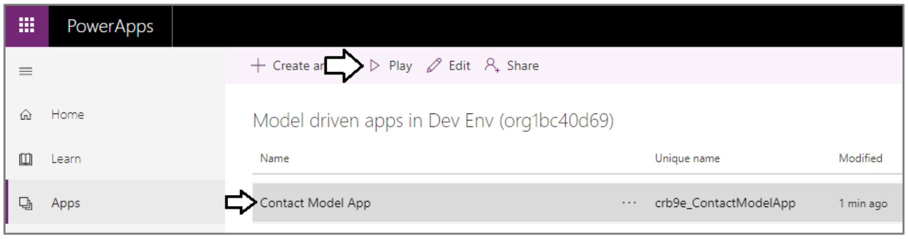
p) After saving the app, click the **Publish** button to make it available for running and testing.



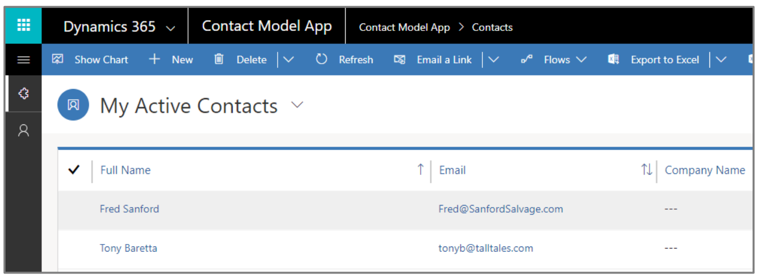
q) Click the **Save and Close** button to close the **App Designer**.



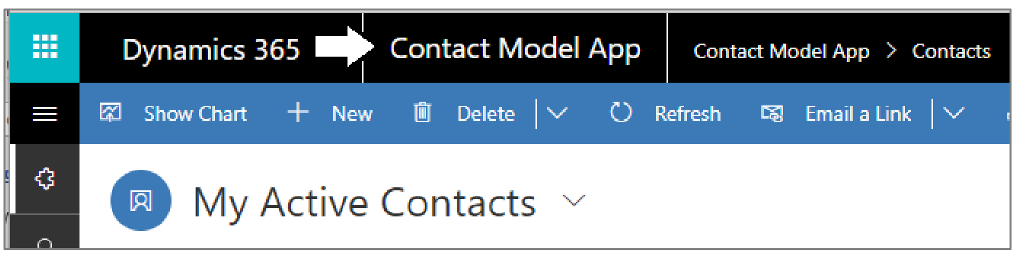
4. Run the model-driven app to test it out.
    a) You should be able to see your new app named **Contact Model App** in the list of **Model driven apps in Dev Env**.
    b) Select the app named **Contact Model App** and then click the **Play** button from the toolbar above.



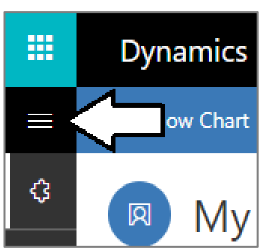
    c) When the app starts, you should see it displays a view named **My Active Contacts**.



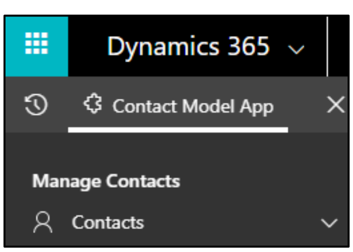
    d) Note the top navigation menu displays the name of your area (i.e. **Contact Model App**) and your subarea (i.e. **Contacts**).



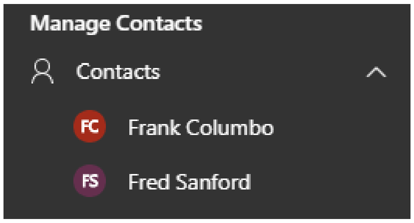
    e) Click the Site Map menu with the 3 horizontal lines in the left navigation to examine the app's site map



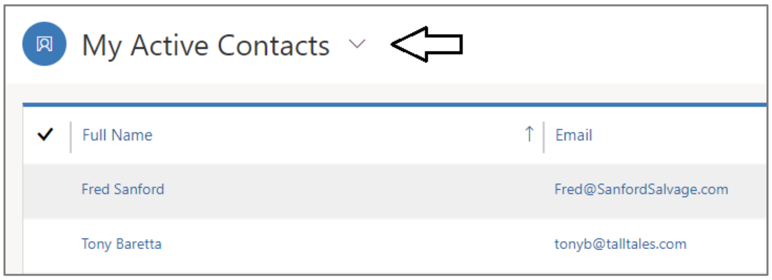
    f) You should now see the site map with your area, group and subarea.



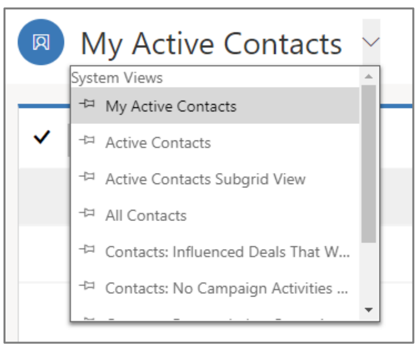
g) If you expand the Contacts subarea, you should see the contacts that you have added to the CDS database.



h) Click on the view menu to the right of **My Active Contacts** to change the view.

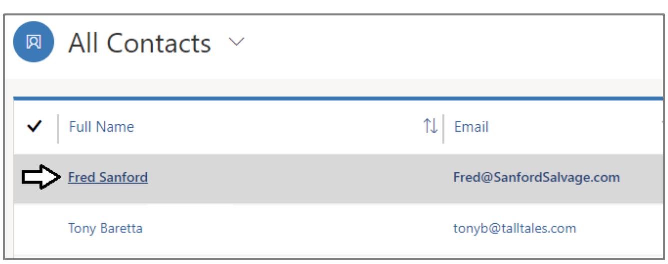


i) When you drop down the view menu, you should see all the views that are include with the standard **Contact** entity.

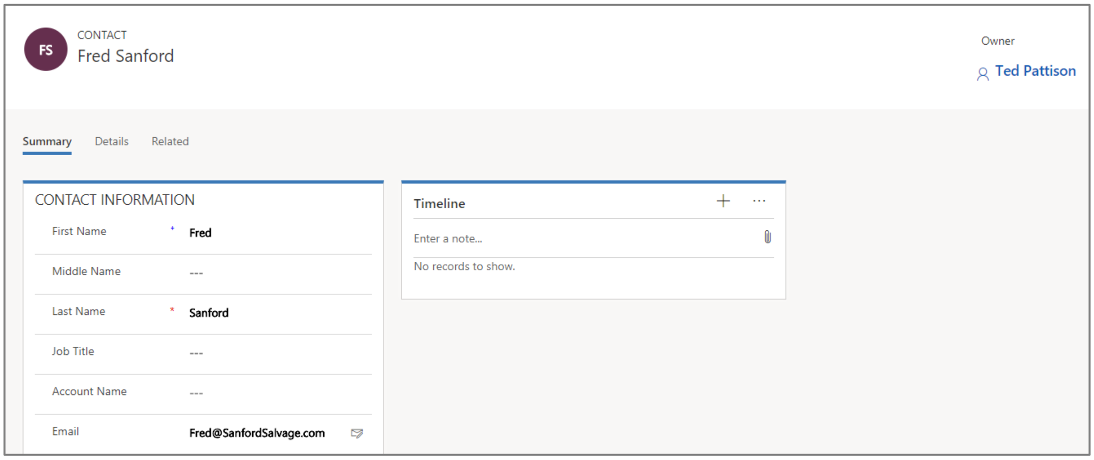


j) Using the view menu, change the view from **My Active Contacts** to **All Contacts**.

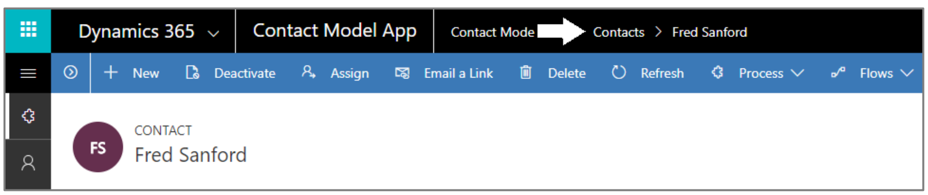
k) In the **All Contacts** view, click on one of the **Full Name** of one of the contacts you created.



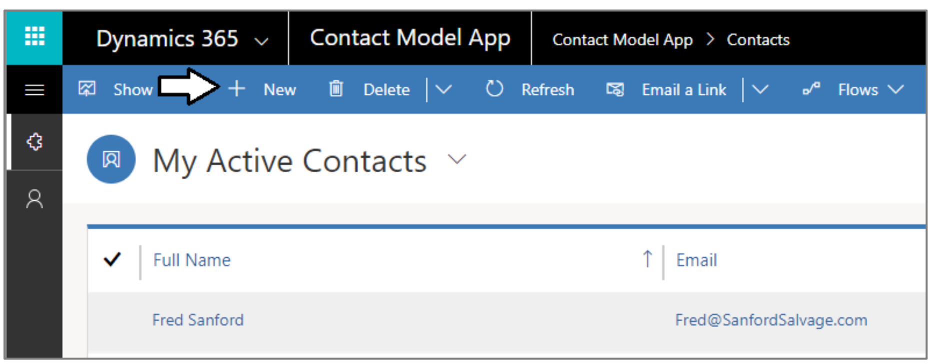
l) You should now see the default form used for the **Contact** entity. Note that you can edit the contact in this view. There is no save command because any edit you make is saved automatically.



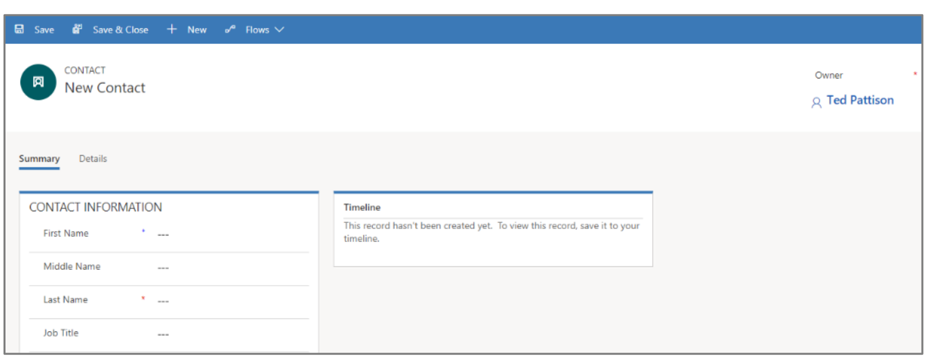
m) Use the top navigation click **Contacts** to move from the form showing one contact back to the My Active Contacts view.



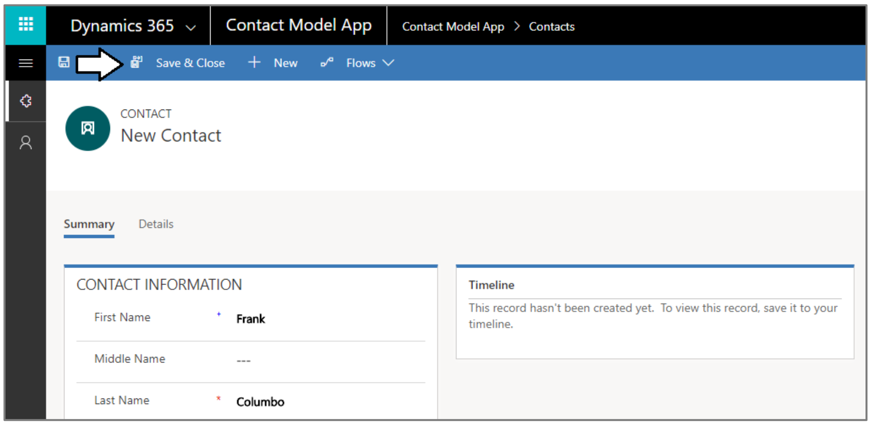
n) Create a new contact by clicking the **+ New** button in the toolbar.



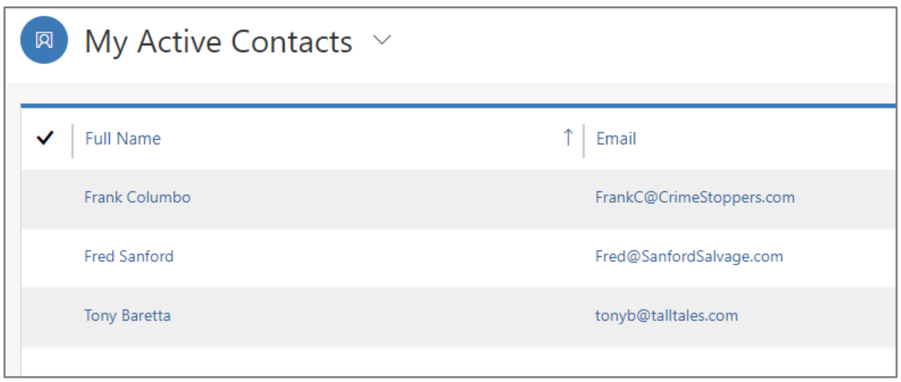
o) You should now see the empty **New Contact** form which allows you to enter the data for a new contact.



p) Enter sample data for another contact and then click the **Save & Close** button.



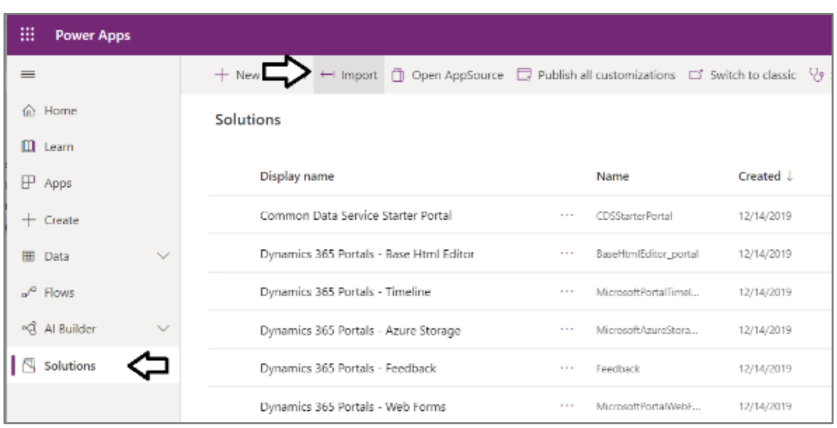
q) You should see that the new contact has been added to the My Active Contacts view.



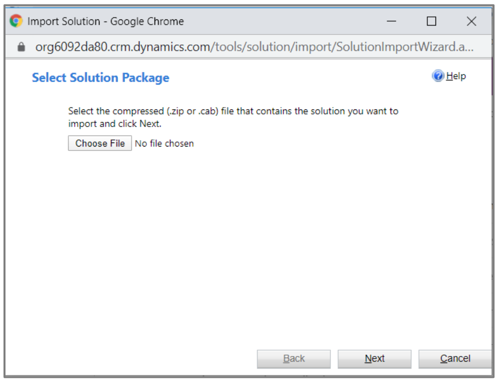
Exercise 5: Import the Product Management Solution

In this exercise, you will import the **Product Management** solution as an unmanaged solution. The **Product Management** solution contains a custom **Product** entity and a model-driven app named **Product Management**. which will make it possible to create an mange product data which will be displayed in your Power Apps portal.

1. Import the **Product Management** solution into **My Dev Environment**.
   a) Make sure the current environment is the environment named **My Dev Environment** that you created in exercise 3.
   b) Click the **Solutions** link in the left navigation and then click the **Import** button on the toolbar.

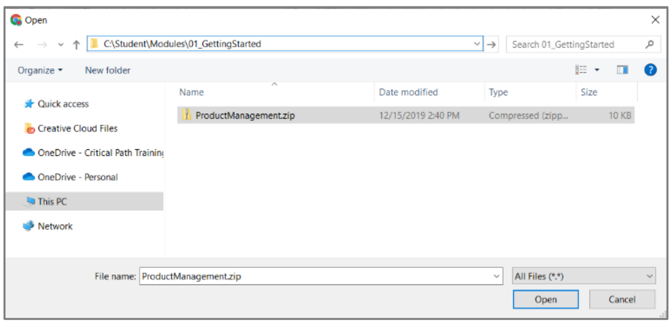


c) You should now see the **Select Solution Package** dialog. Click the **Choose File** button.

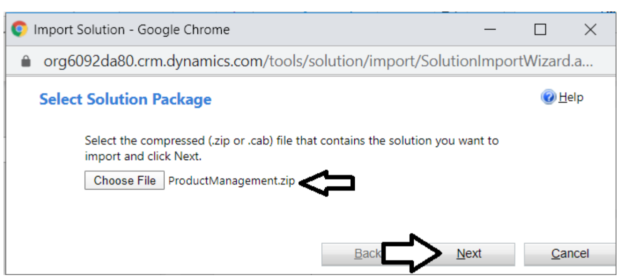


d) Select the file from inside your **Student** folder at the following path and then click **Open**.

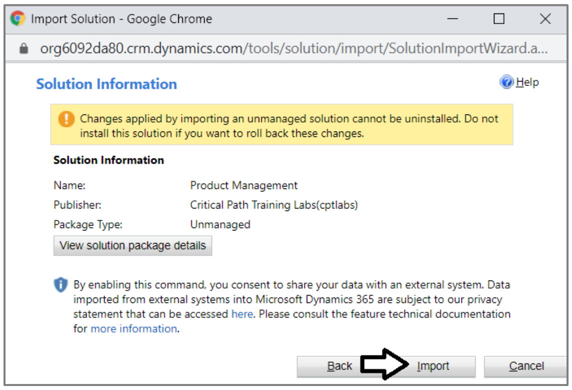
```
C:\Student\Modules\07_CommonDataService\Lab\ProductManagement.zip
```



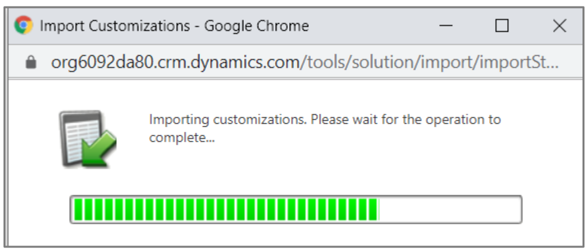
e) Once the file name of **ProductManagement.zip** appears next to the **Choose File** button, click **Next**.



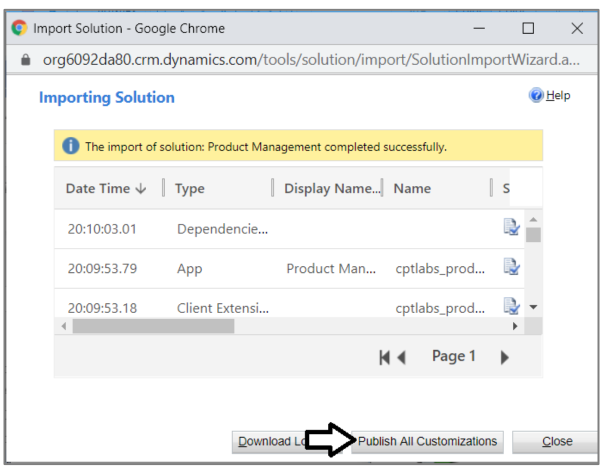
f) Review the information on the **Solution Information** dialog and then click **Import** to import the solution.



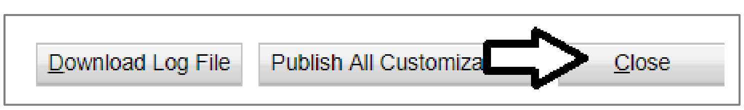
g) Wait while the solution package is imported.



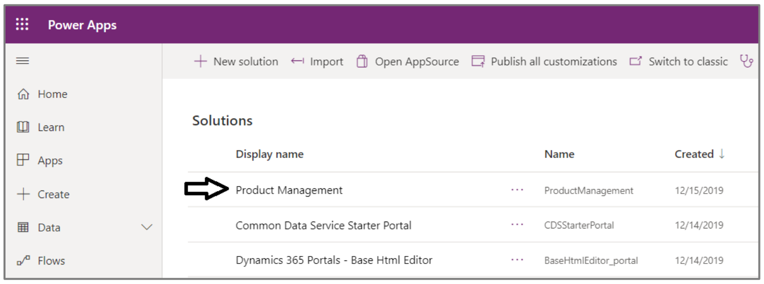
h) After the solution has been imported, click **Publish All Customizations**.



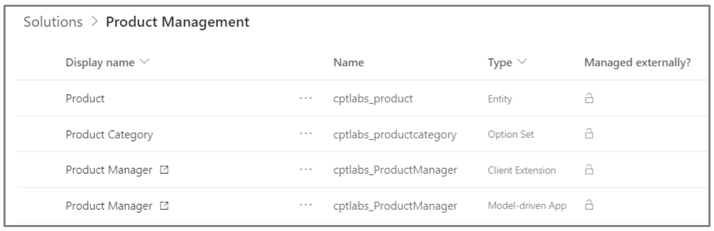
i) After you have published al customization, click **Close** to close the **Importing Solution** dialog.



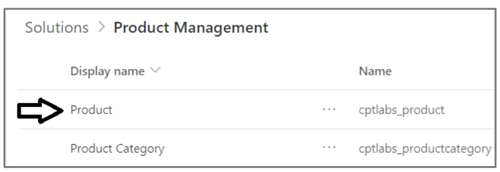
j) After import the **Product Management** solution, locate it in the **Solutions** list for your environment and click on it.



k) You should be able to see that the **Product Management** solution contains the **Product** entity and three other components.

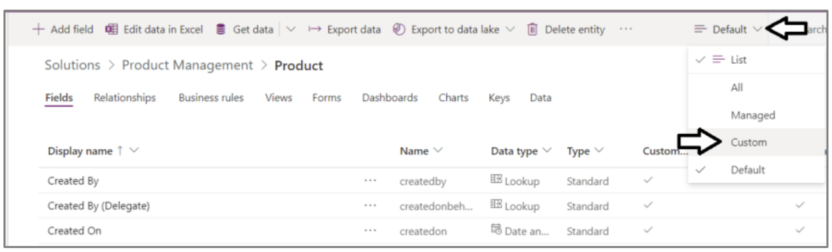


l) Double click on the **Product** entity to see its definition.

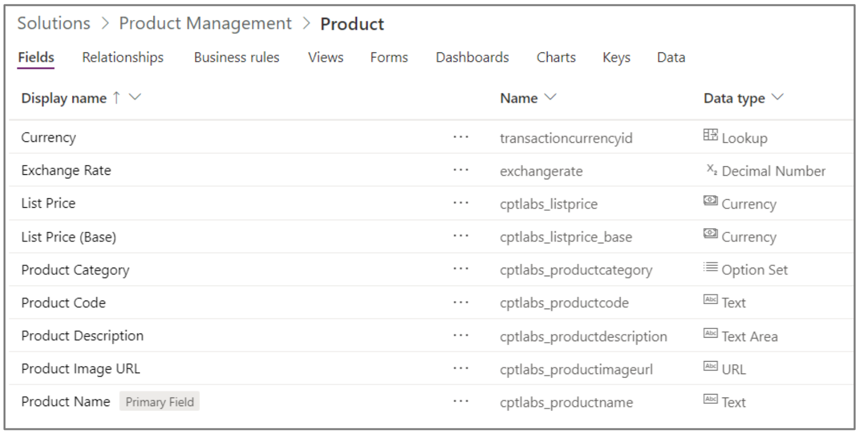


m) You should see the **Fields** for the **Product** entity.

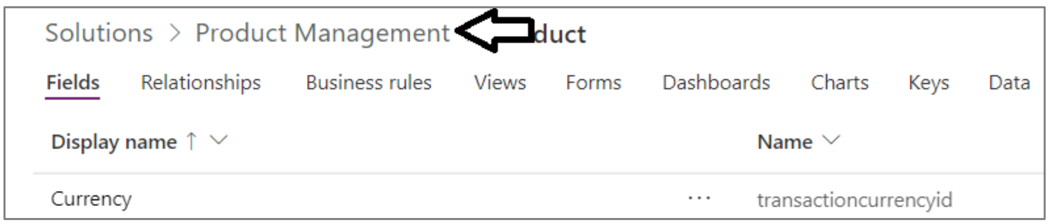
n) Drop down the filter menu on the right and set the filter to **Custom** to only display fields that are custom fields.



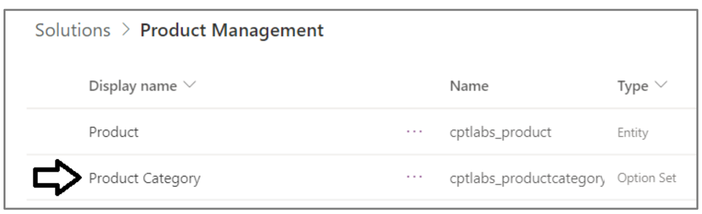
o) You should now see the custom fields that make up the **Product** entity.



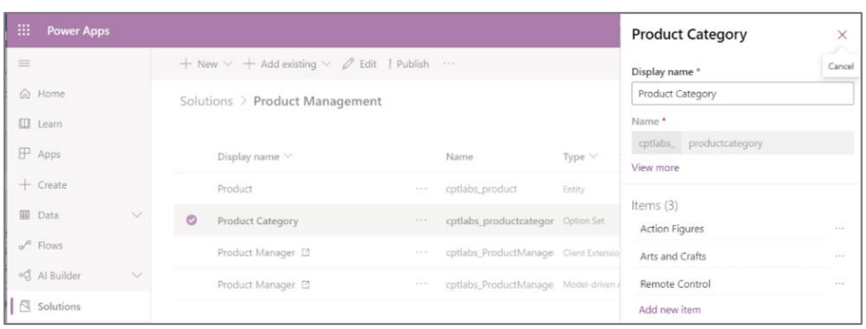
p) On the breadcrumb menu, select **Product Management** to move back to the **Product Management** solution.



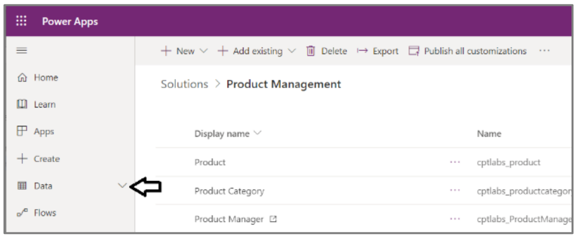
q) Click on the **Product Category** option set to see the possible values for the **Product Category** column.



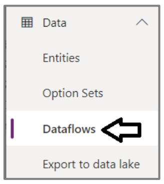
r) You should see the **Product Category** option set values are **Action Figures**, **Arts and Crafts** and **Remote Control**.



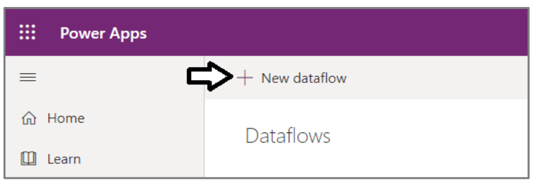
2. Create a dataflow named **Import Products** to import product data into the Product entity recordset.
   a) Expand the **Data** node in the left navigation.



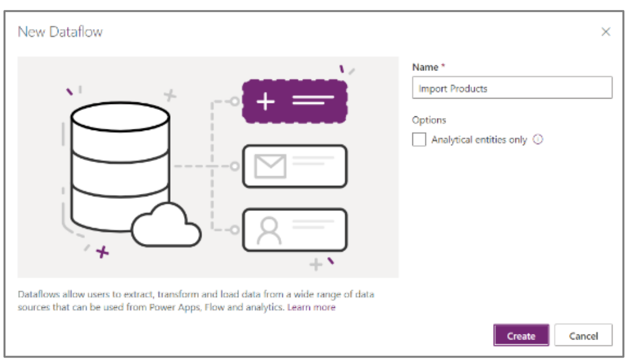
b) Inside the **Data** node, click **Dataflows**.



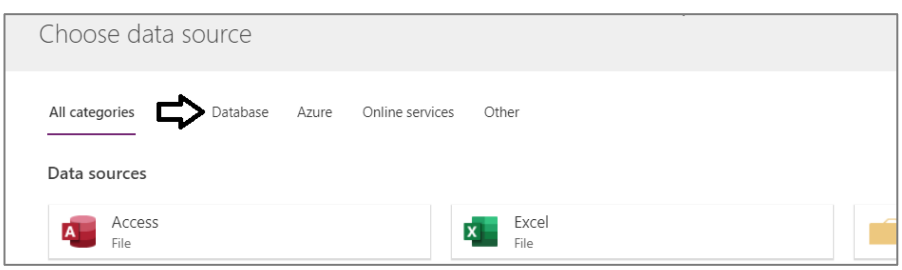
c) Click **+ New Dataflow**.



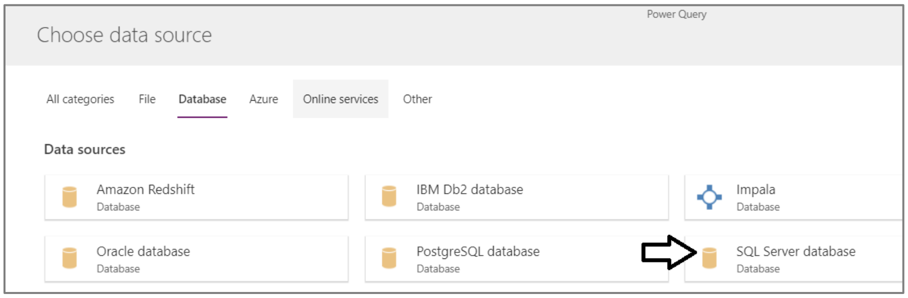
d) In the **New Dataflow** dialog, enter a name of **Import Products** and then click **Create**.



e) On the **Choose data source** page, click the **Database** tab.

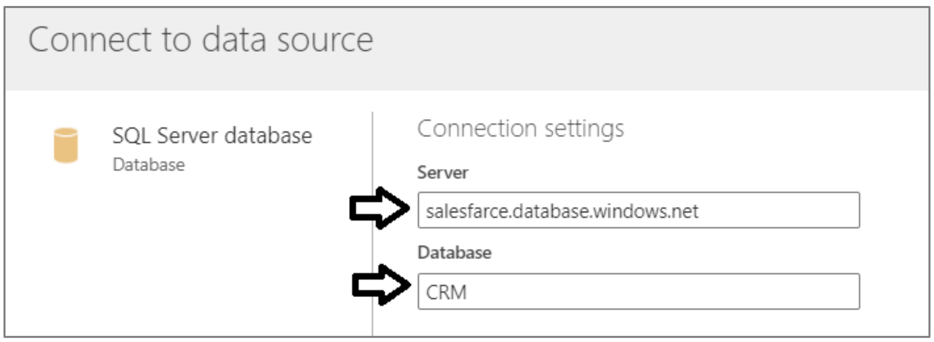


f) Click on **SQL Server database**.



g) Enter for the **Server** setting.

h) Enter **CRM** for the **Database** setting.

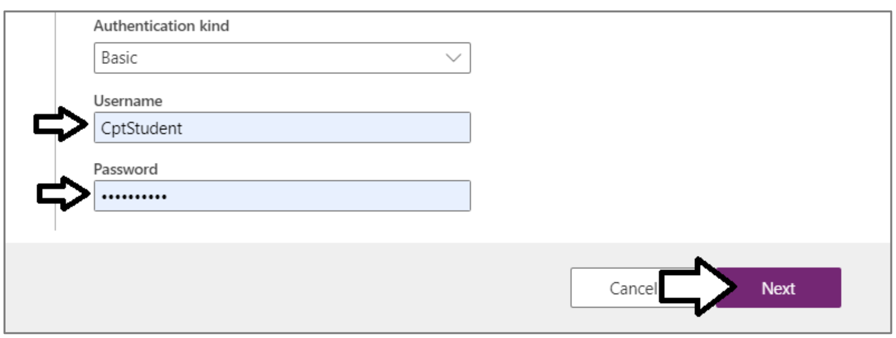


i) Leave the **Authentication kind** setting with the default value of **Basic**.

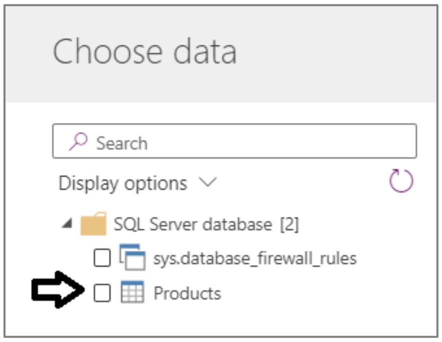
j) Enter a **Username** of **CptStudent**.

k) Enter a **Password** of **pass@word1**.

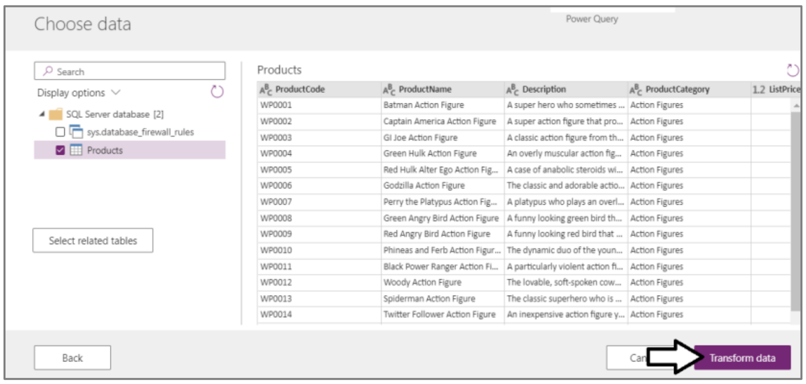
l) Click **Next**.



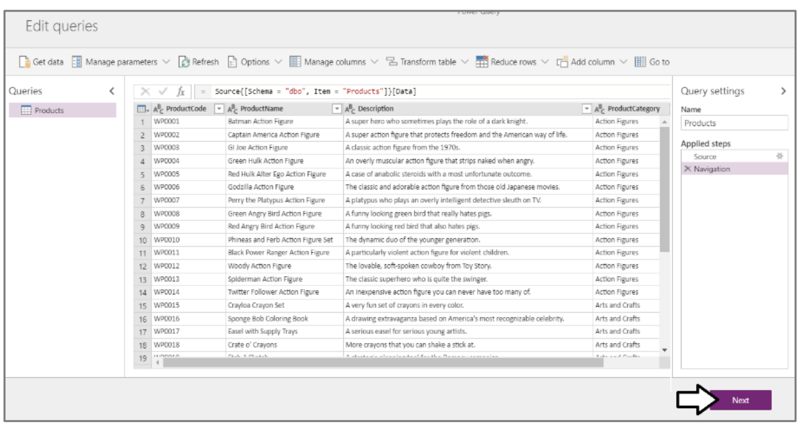
m) When prompted to **Choose data**, select the **Products** table.



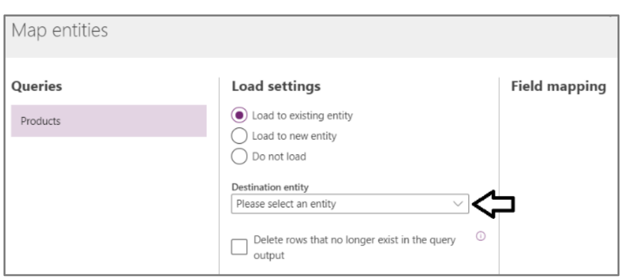
n) Click the **Transform data** button in the lower right corner of the screen.



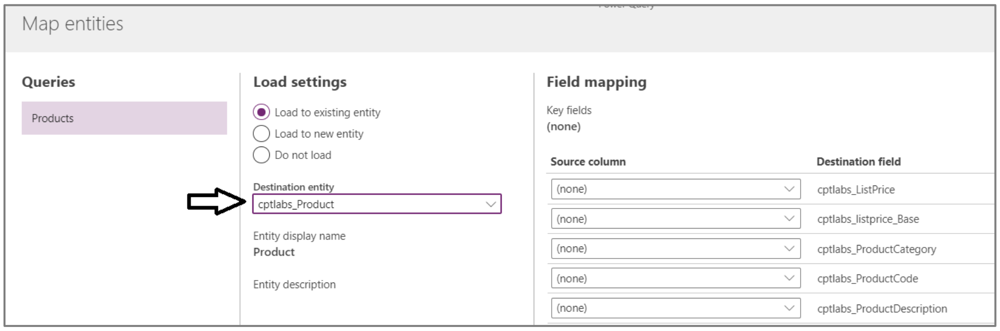
o) Click **Next**.



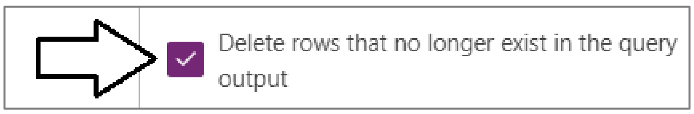
p) On the **Map entities** screen, select **Load to existing entity** and drop down the **Destination entity** menu.



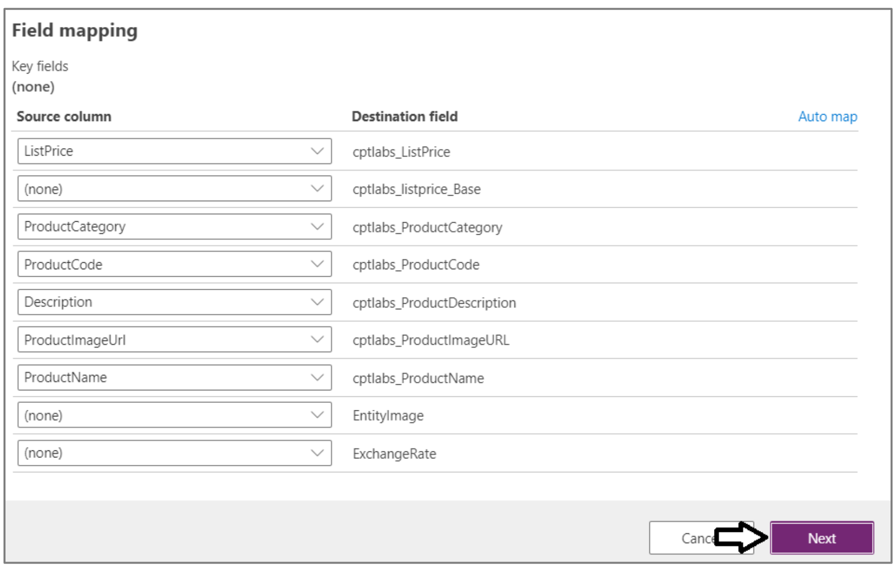
q) Select a **Destination entity** setting of **cptlabs_Product**.



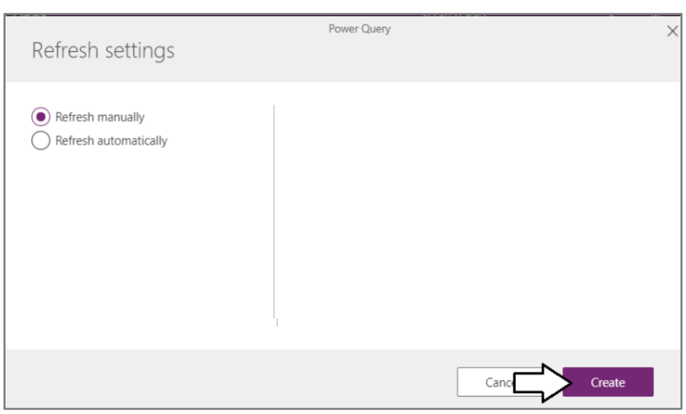
r) Select the setting **Delete rows that no longer exist in the query output**.



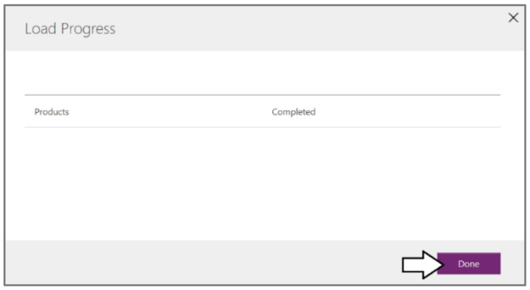
s) In the **Field mapping** pane, map all fields as shown in the following screenshot and then click **Next**.



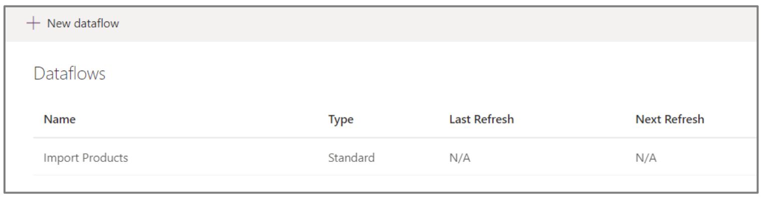
t) In the **Refresh setting** dialog, select **Refresh manually** and then click **Create**.



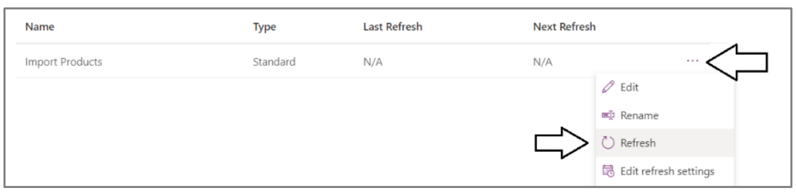
u) You should see a Load Progress dialog that indicates the load has completed.



v) You should now see the **Import Products** dataflow in the **Dataflows** list.



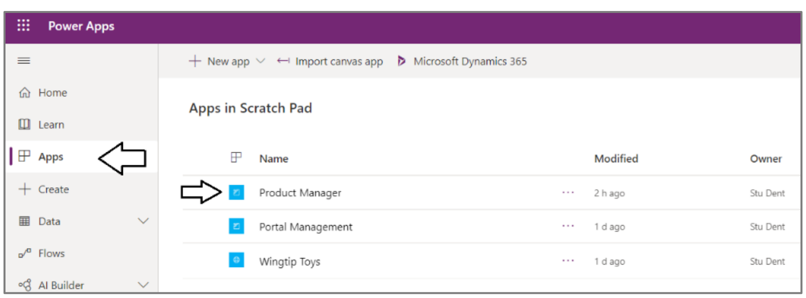
w) Note that you can drop down the ellipse menu and run the **Refresh** command to run the dataflow and refresh the data.



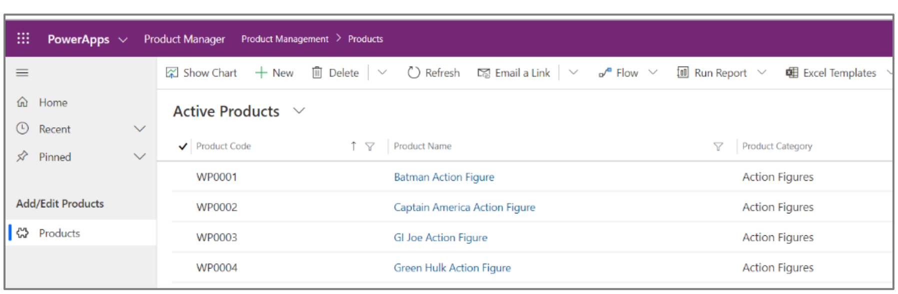
3. Run the model-driven app named Product Manager to view and enter product data.

a) Click the **Apps** link in the left navigation.

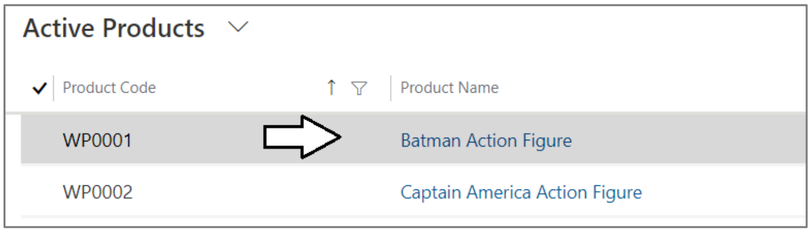
b) Click **Product Manager** to launch this model-driven app.



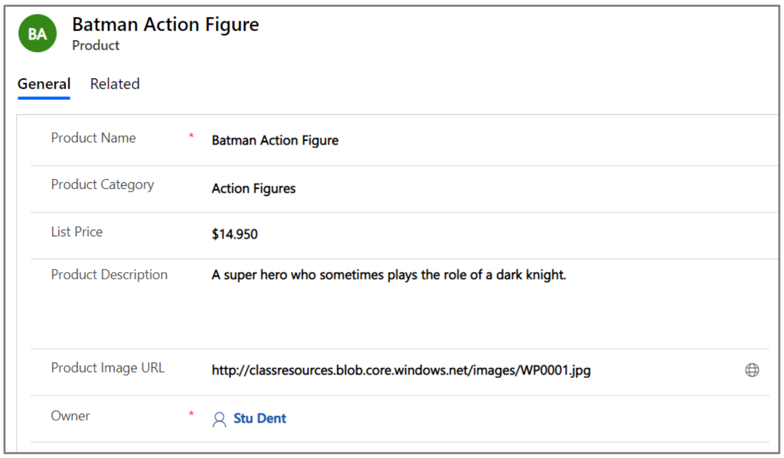
c) You should now see a view which displays all products.



d) Double click on the top product named Batman Action Figure to display the form for this record.

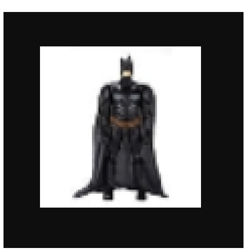


e) You can now see the values for reach field in this **Product** record



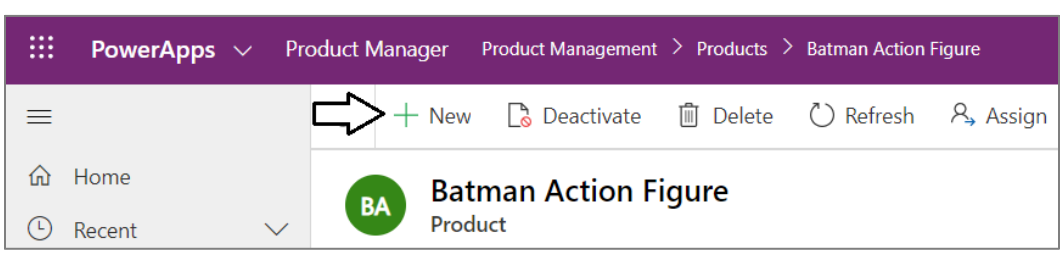
f) Click the globe icon to the right of the **Product Image URL** to display that URL in browser tab.

g) You should see the image for this product in a new browser tab. Close this browser tab once you have seen the picture.

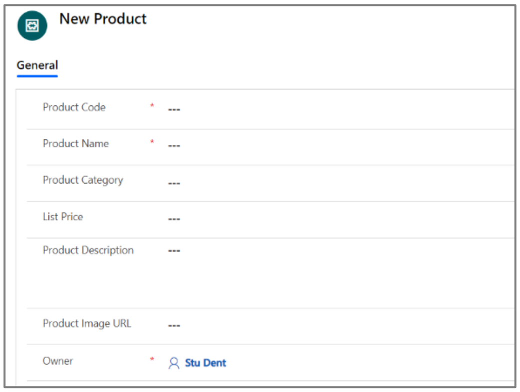


4. Create a new Product record.

a) Click the **+ New** button to display the **New Product** form.

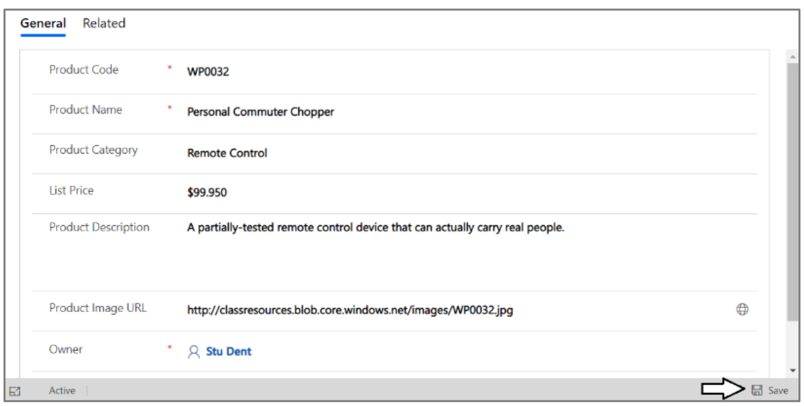


b) The **New Product** form should match the following screenshot.

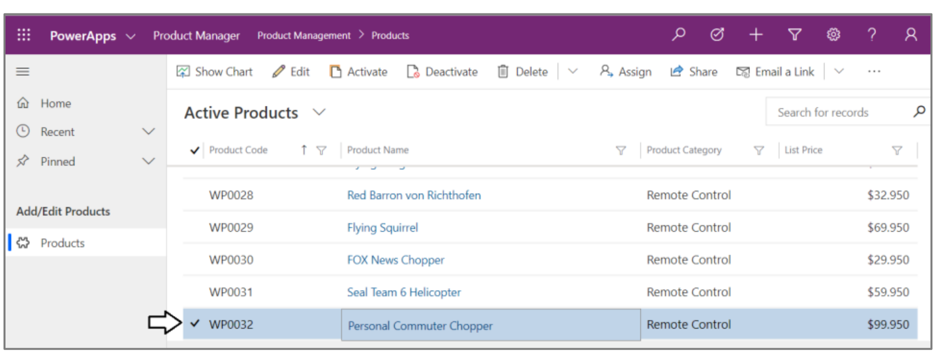


c) Fill in the **New Product** form with the following data.
   i) **Product Code**: WP0032
   ii) **Product Name**: Personal Commuter Chopper
   iii) **Product Category**: Remote Control
   iv) **List Price**: $99.95
   v) **Product Description**: A partially-tested remote control device that can actually carry real people.
   vi) **Product Image URL**: http://classresources.blob.core.windows.net/images/WP0032.jpg

d) When your form matches the following screenshot, click the **Save** button to save your changes.



e) Return to the Active Products list and verify you can see the new record.



Congratulations. You have just created and tested a simple model-drive app. You have now reach the end of this lab.