

Power of the Cloud

January 29th, 2021

8:30am - 4:30pm EST



Power of the Cloud Conference 2021

<https://powerofthecloud.net/>

Table of Contents

Lab Pre-Requisites.....	2
Step 1: Create a Development Tenant.....	2
Step 2: Create a SharePoint Site	2
Step 3: Create a SharePoint List.....	3
Lab 1 – Create an App	8
Connect to data.....	9
Using global variables, Collect and ClearCollect	11
Syntax.....	13
Syntax.....	14
New screen (show results).....	14
Using Filter (filtering results)	18
Syntax.....	18
Using Distinct	19
Syntax.....	19
Using Search.....	20
Syntax.....	20
Using SortByColumns.....	20
Adding new Items	22
Remove Items	24
Lab 2 - Create an approval workflow triggered by the Power App	26
Lab 3 - Building the Population Report – Data Prep	44
Building the Population Report – Visuals.....	50
Embedding in SharePoint.....	55

Lab Pre-Requisites

Step 1: Create a Development Tenant

If you have not created a development tenant, please do so by following the steps in this video:

[How to Setup a Trial Microsoft 365 Environment to use Power Platform, Power BI & SharePoint Syntex - YouTube](#)

Step 2: Download Power BI Desktop

Download and install the Power BI Desktop: <https://www.microsoft.com/en-us/download/confirmation.aspx?id=58494>

Step 3: Create a SharePoint Site

1. Login to your tenant
2. Navigate to <https://www.office.com>
3. Click on the SharePoint App



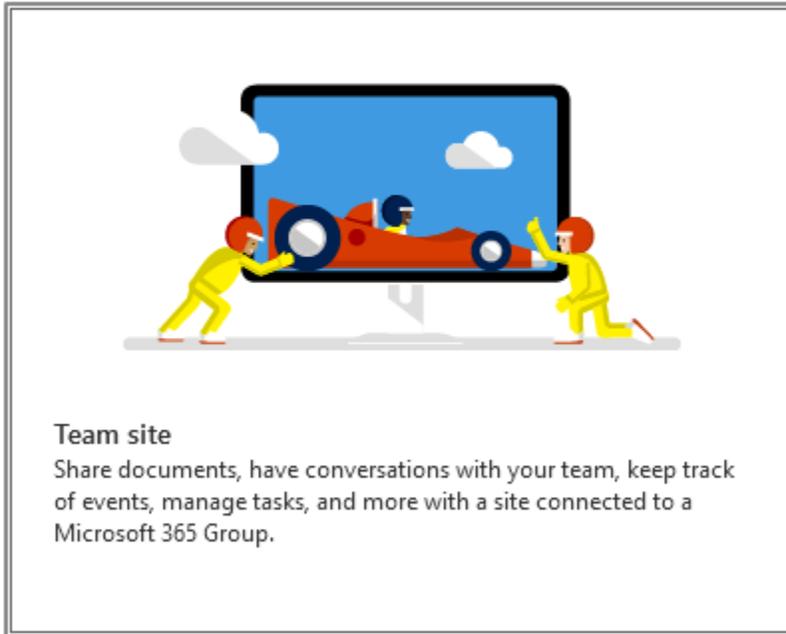
4. Click on **Create site**

+ Create site + Create news post

5. Select Team Site

Create a site

Choose the type of site you'd like to create.



6. Call it Power of the Cloud
7. Click **Next**
8. Click **Finish**

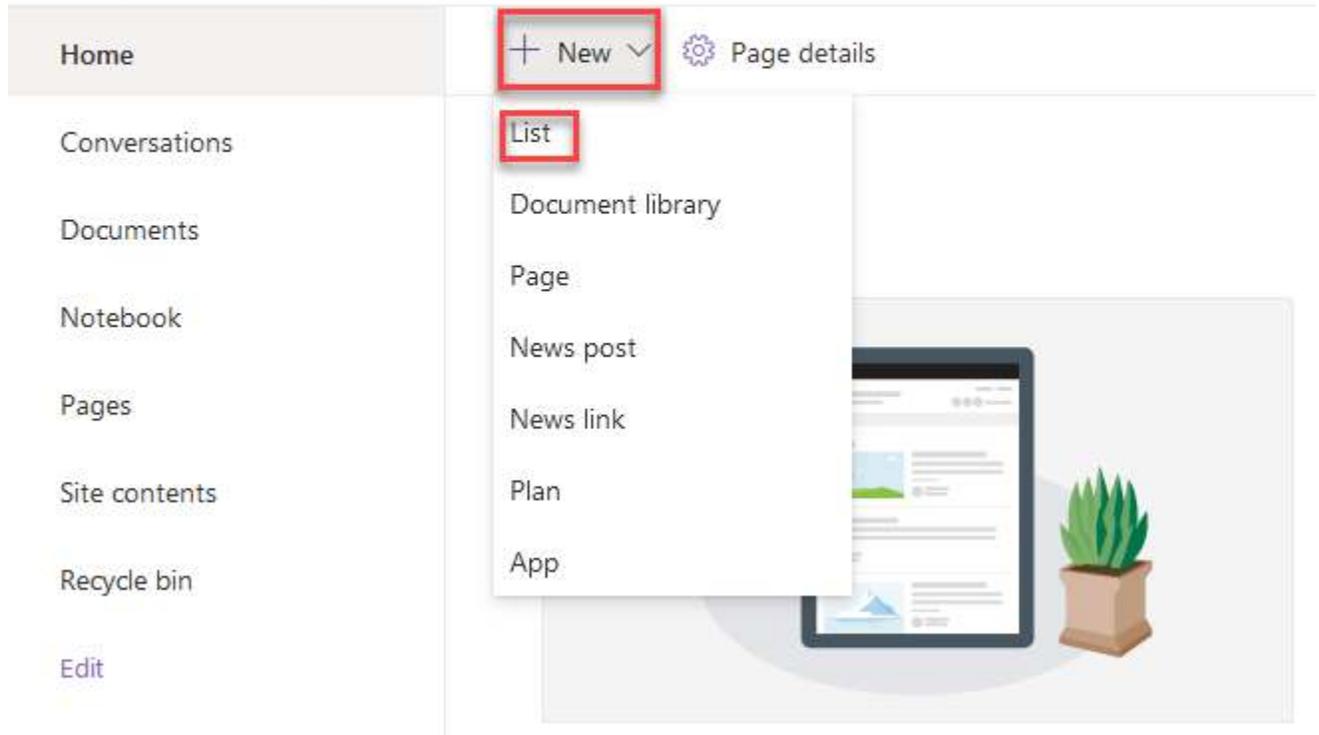
Step 4: Create a SharePoint List

1. Download the spreadsheet from: <https://powerofthecloud.net/wp-content/uploads/2021/01/Tutorial-Data.xlsx>
2. In your new SharePoint site, Click on New → List



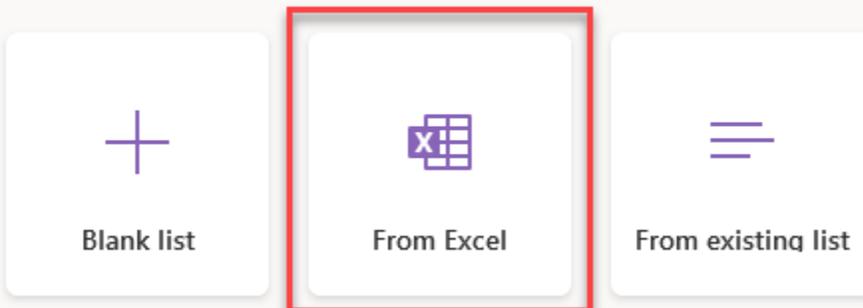
Power of the Cloud

Private group



3. Select **From Excel**

Create a list



4. Click on **Upload file**

From Excel

Select an Excel file from your device or this site.

Upload from this device

Upload file

Choose a file already on this site

Documents

 Name	Modified	Modified by
--	----------	-------------

5. Select the Tutorial Data.xlsx that you downloaded
6. Click on **Next**
7. Call it **City Popluations**

Name * ✕
City Populations

Description

What is your list about?

Site navigation

Show in site navigation

[< Back](#)

Create

Cancel

8. Click **Create**
9. Click on **Add Column** → Select **Yes/No**

City Populations

Title ▾	Country ▾	Population ▾	+ Add column ▾
📍 Cairo	Egypt	9,000,000	Single line of text
📍 Rome	Italy	2,873,000	Multiple lines of text
📍 Milan	Italy	1,352,000	Location
📍 Moscow	Russia	12,538,000	Number
📍 Istanbul	Turkey	15,190,000	Yes/No
📍 Mexico City	Mexico	21,782,000	Person
📍 Rio De Janeiro	Brazil	13,458,000	Date and time
📍 Johannesburg	South Africa	5,783,000	Choice
📍 Nairobi	Kenya	4,735,000	Hyperlink
📍 Ottawa	United States	500,000	Currency
📍 Ottawa	Canada	1,000,000	Image
			More...
			Show/hide columns

10. Call it **ApprovalStatus** with the default set to **No**

Create a column



[Learn more about column creation.](#)

Name *

Description

Type

Default value

More options

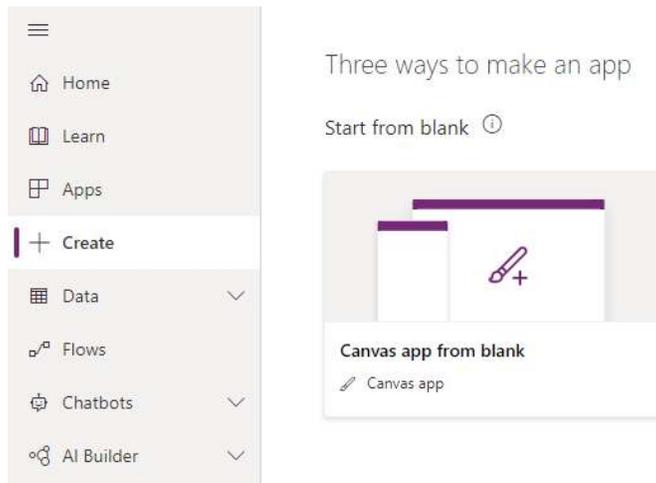
11. Click **Save**

Congratulations! You have now completed the pre-requisites to start your lab!

Proceed to the next page for your first Lab.

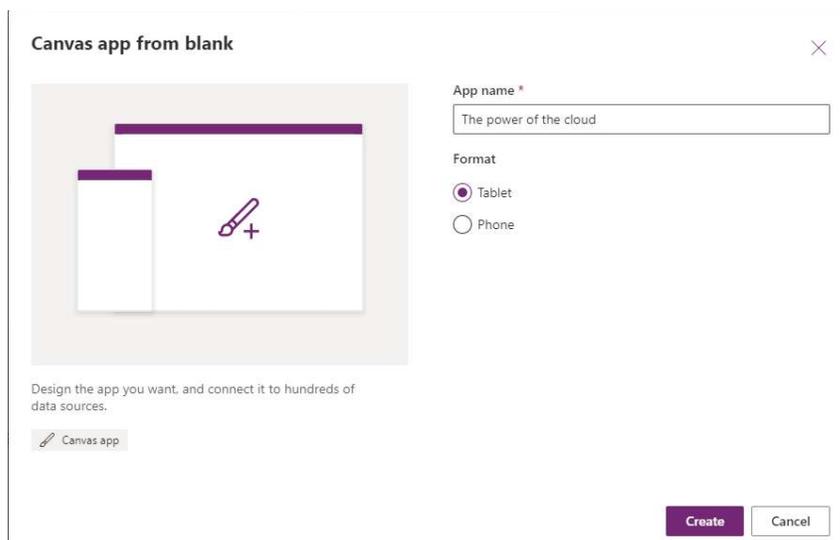
Lab 1 – Create an App

1. Sign in to [Power Apps](#).
2. Select **Create** from left navigation. Select **Canvas app from blank** drop down menu and then select **Canvas**.



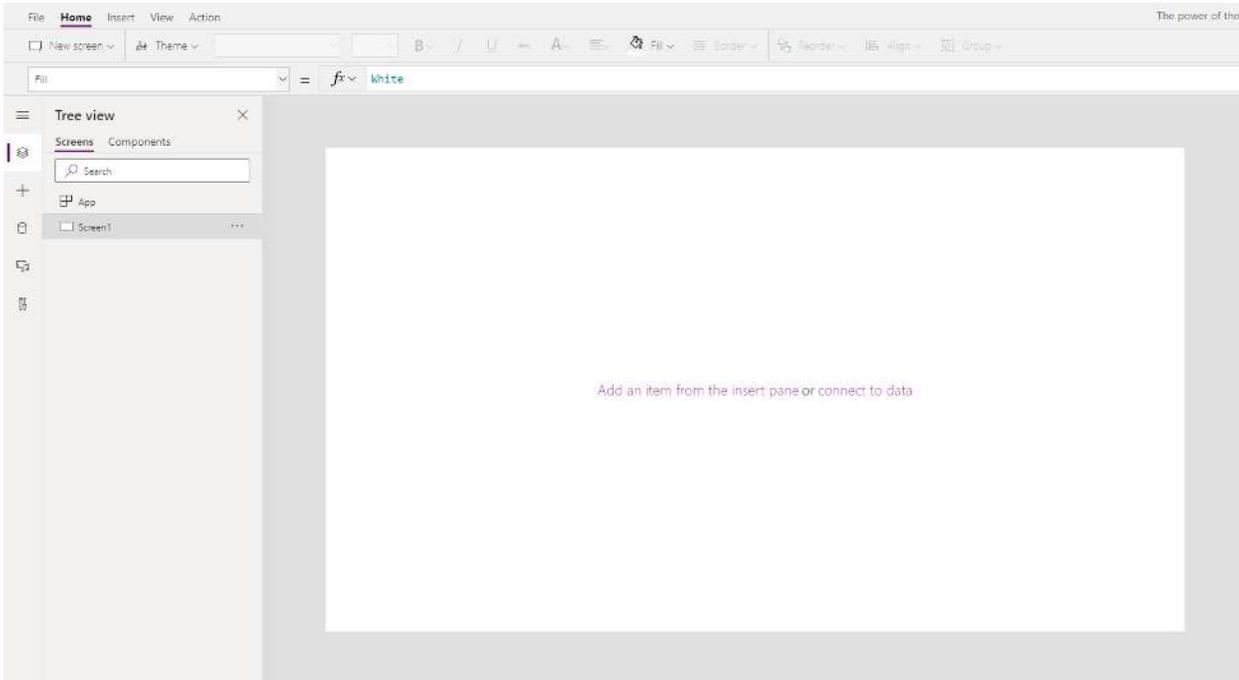
This opens a popup.

3. In the **App name** write down a name for your app (this can be change later "if you like"), keep **Tablet** selected in the **Format** section and click **Create**.



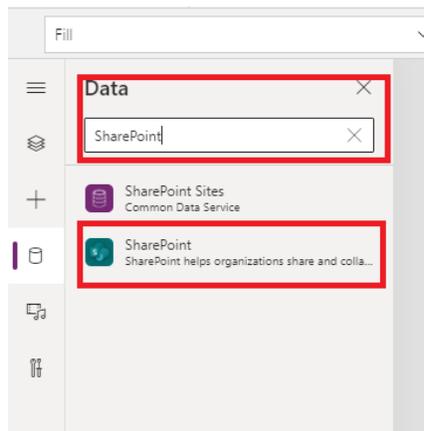
This will open in a new browser tab.

4. This is what you should get after the new browser tab is loaded.
5. Under the **Tree View** tab (left side) is where the controls you drop on the page will be listed.

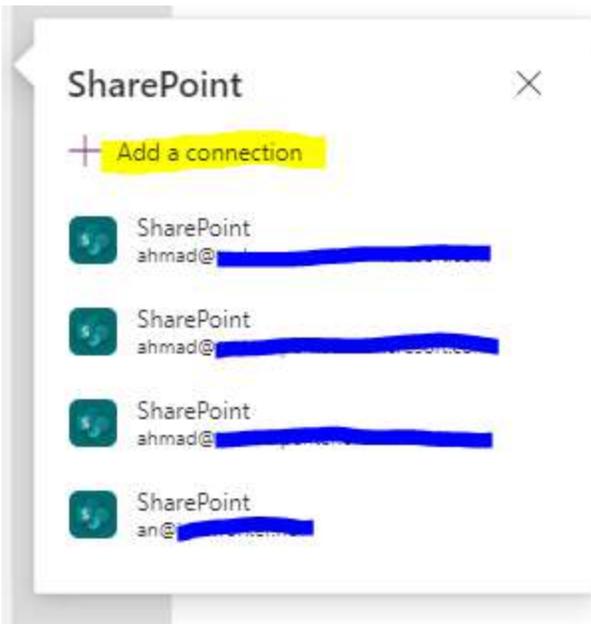


Connect to data

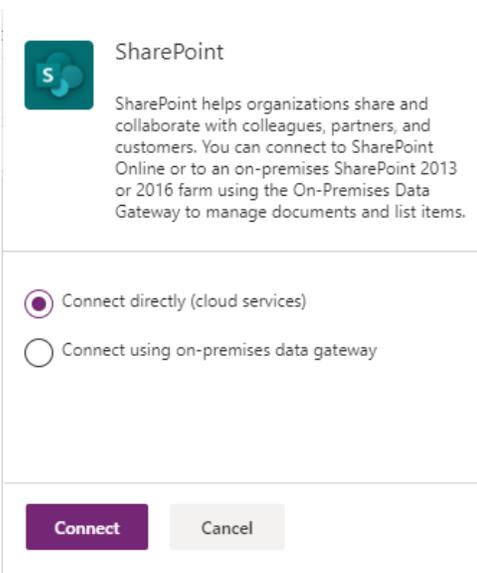
1. Let us connect to our data source first (Population – SharePoint list) “if you haven’t created this list in SharePoint, make sure that you refer to the documentation on how to create your SharePoint list”.
2. Click on the **data** icon on the left menu from the left.
3. Under **Data**, you will find the different data sources that you can connect to in Power Apps. Write down **SharePoint** in the search box and then select **SharePoint**.



4. Once you click on the SharePoint connector, you will see a list of connections that you have created from before. Since this is your first app, click on add connection to connect to a SharePoint site where your list resides.



5. This will open a panel on the right-hand side.
6. Choose "connect directly (cloud services)" and then hit **connect**.



7. Under **Connect to a SharePoint site**, type or paste the URL for your SharePoint Online site, and then select **Connect**.

← Connect to a SharePoint site ×

Enter the SharePoint URL for the location of your list.
[Learn more](#)

https://my.sharepoint.com/sites/ThePowerOfTheClo...

Connect

- Under **Choose a list**, select **population list**, or **search** for the list using the search textbox on the top.

← Choose a list ×

Search

AI Builder

Bank public news content approval

Documents

Language detection

News content approval

Population

Sentiment analysis

Text with Categories

Text with Entities

Vacation approval

Enter custom table name

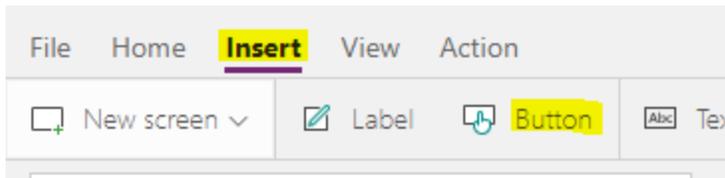
Connect Cancel

- The list should now appear under **In your app** data.

Using global variables, Collect and ClearCollect

For this exercise we need to drag and drop a couple of controls.

- From the top menu. Choose **Insert** and then click on **Button**. We need 3 buttons for this exercise, so make sure you have 3 buttons in your screen.

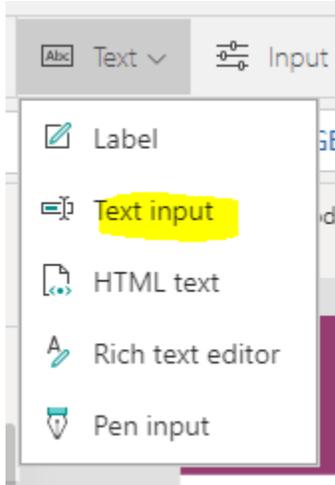


Set Variable

Collect

ClearCollect

2. Now, from **Insert** tab, click on **Text** and then choose **Text input**.



Your screen should look something like this 😊



You can change the text on your button(s) from the properties right pane (when the button is selected).

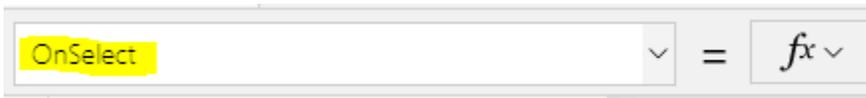
3. We want to set a value for your global variable by using the following syntax:

Syntax

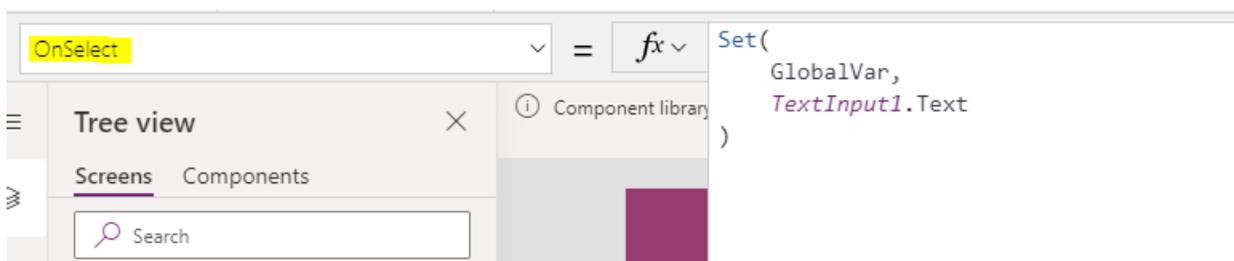
Set(*VariableName*, *Value*)

- *VariableName* - Required. The name of a global variable to create or update.
- *Value* - Required. The value to assign to the context variable.

4. Make sure that you select the button and the selected event for the button is **OnSelect**.



5. We need to set the value of whatever text we have in the **text input** to the global variable; accordingly, your syntax should look like this:



6. We need to do something similar for both **Collect** and **ClearCollect** buttons.

7. The Collect and ClearCollect syntax looks like this

Syntax

Collect(*DataSource*, *Item*, ...)

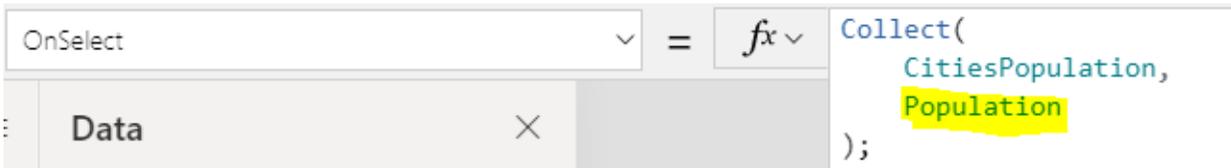
- *DataSource* – Required. The data source that you want to add data to. If it doesn't already exist, a new collection is created.
- *Item(s)* - Required. One or more records or tables to add to the data source.

ClearCollect(*Collection*, *Item*, ...)

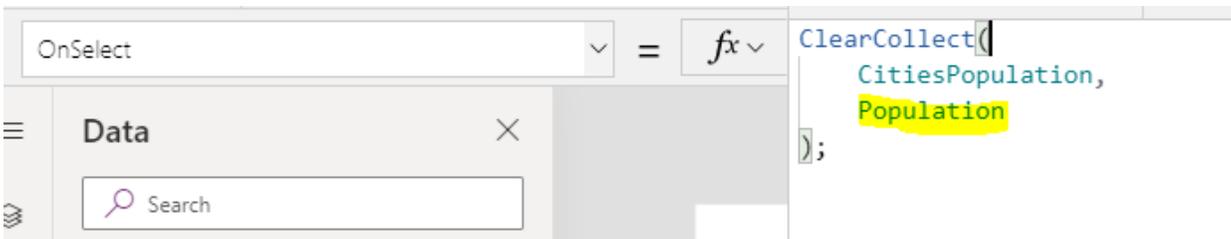
- *Collection* – Required. The collection that you want to clear and then add data to.
- *Item(s)* - Required. One or more records or tables to add to the data source.

8. **Collection** is the name of our new (global) data collection and **Item(s)** is our data source (**SharePoint list**) **highlighted in the snapshot below.**

9. Make sure that the **Collect** button is selected and the selected event for the button is **OnSelect**

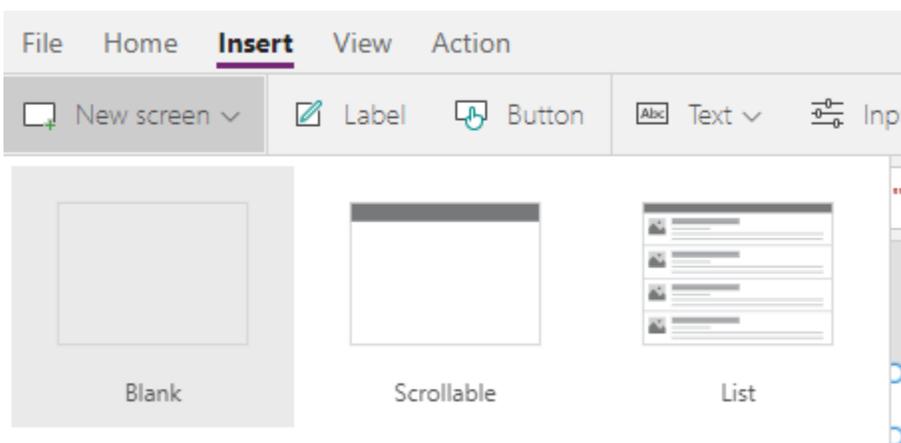


10. Make sure that the **ClearCollect** button is selected and the selected event for the button is **OnSelect**

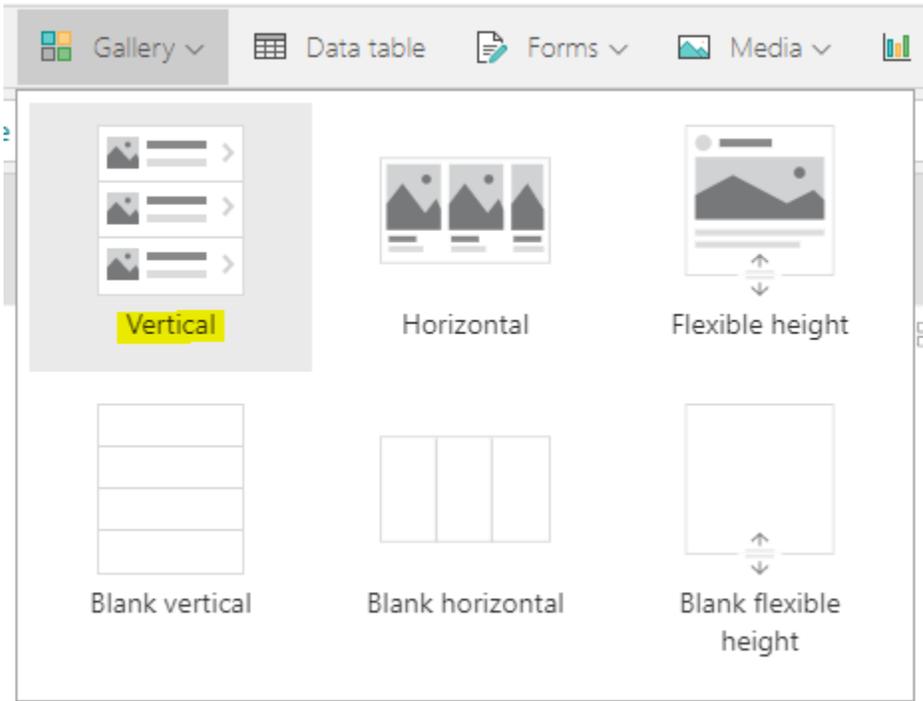
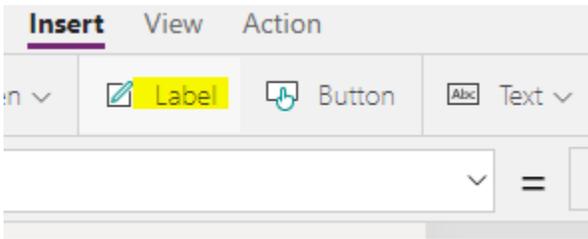


New screen (show results)

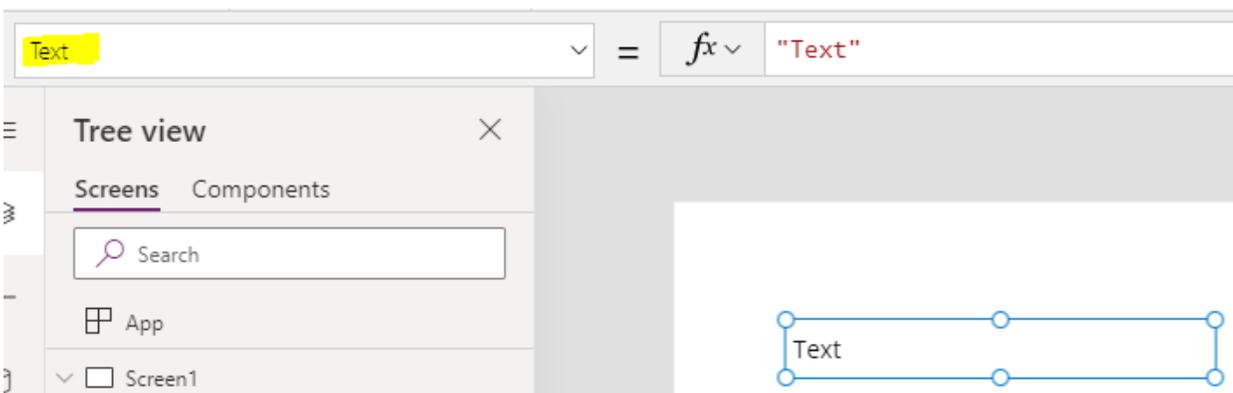
1. From the upper menu choose **New screen** and then select **Blank**



2. In the new screen insert a **Label** and a **Vertical** gallery



3. Make sure that the **Label** is selected and the selected property is **Text**

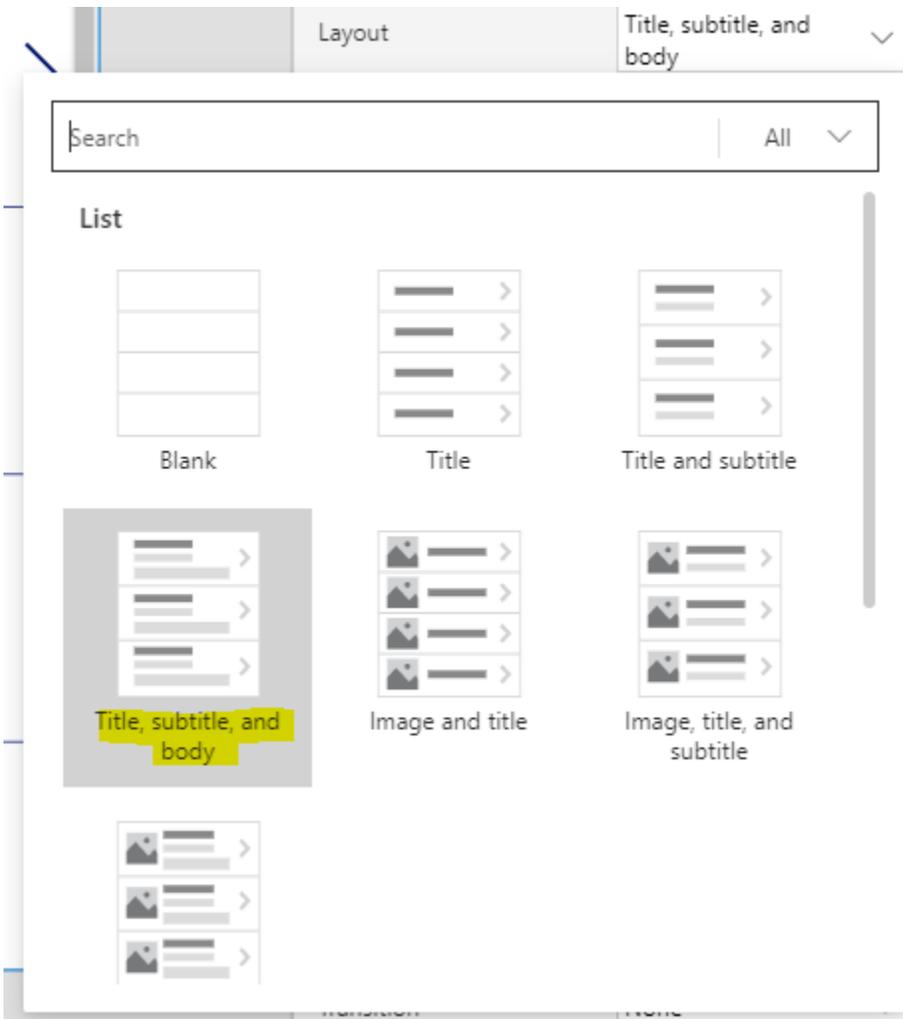


4. Change the text value to the name of your global variable



5. Before binding the data to our gallery, we want to change the layout from the right property pane

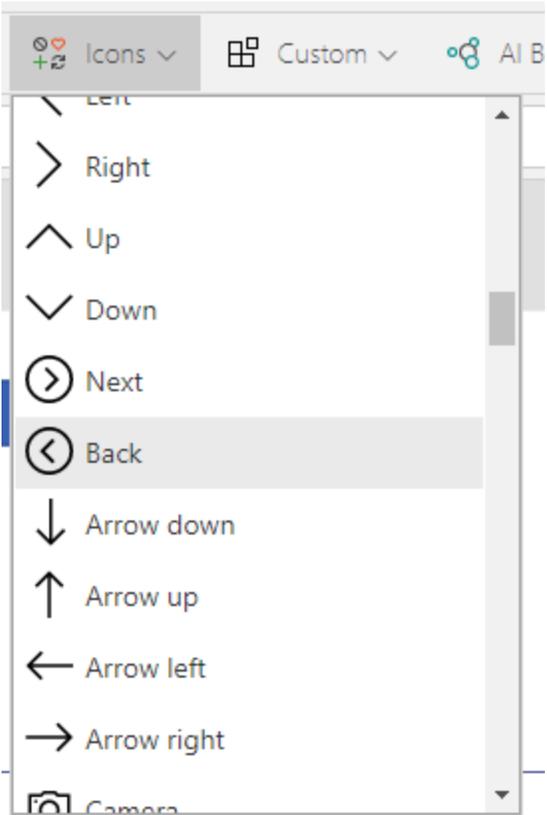
6. Make sure that the gallery is selected and then from the right property pane choose layout and select (title, subtitle and body)



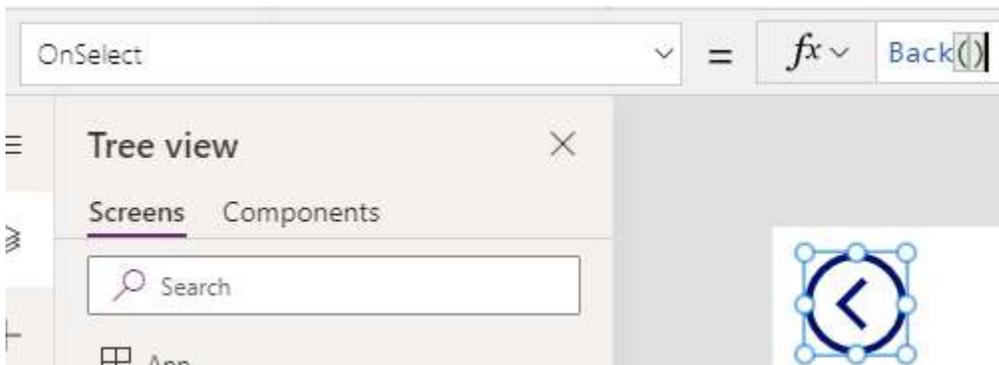
7. Make sure that the gallery is still selected and change the **Items** property to our global collection variable.



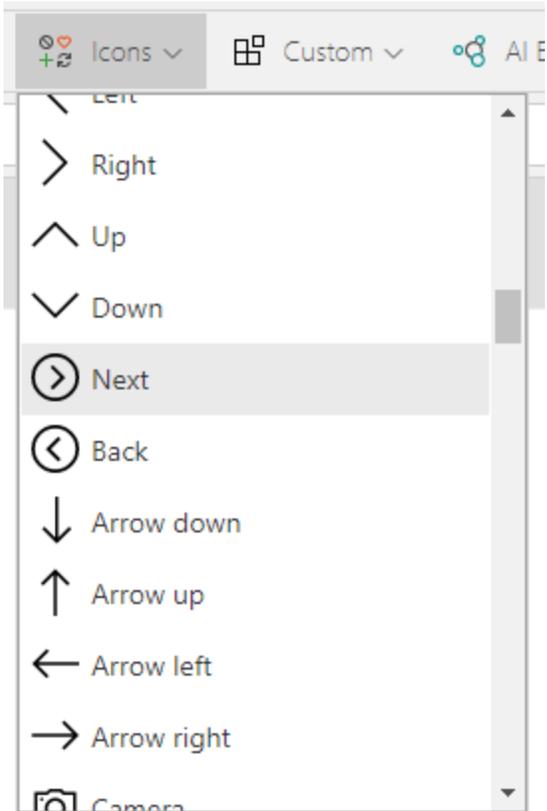
8. To be able to navigate between screen Insert a back icon



9. Make sure that the back icon is still selected and in the formula bar write Back()



10. From the tree view, go back to the first screen and insert a next icon



11. Make sure that the next icon is still selected and in the formula bar write **Navigate(Screen2)**

Using Filter (filtering results)

To filter results, use the following syntax

Syntax

Filter(*Table*, *Formula1* [, *Formula2*, ...])

- *Table* - Required. Table to search.
- *Formula(s)* - Required. The formula by which each record of the table is evaluated. The function returns all records that result in **true**. You can reference columns within the table. If you supply more than one formula, the results of all formulas are combined with the **And** function.

In this exercise we will filter cities population by country, to do that change the formula in the Items property to:

```
Filter(CitiesPopulation, Country = "Norway")
```

This will only show cities in Norway.

Using Distinct

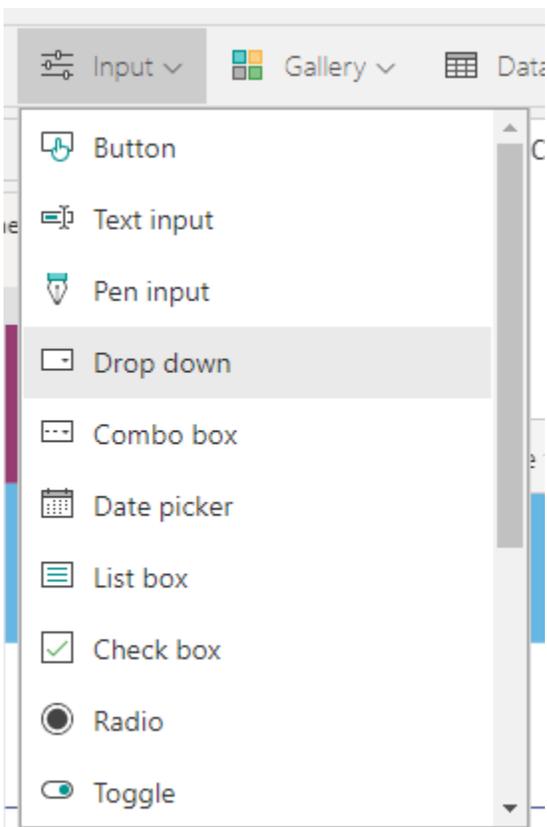
The **Distinct** function evaluates a formula across each record of a table and returns a one-column table of the results with duplicate values removed. The name of the column is **Result**.

Syntax

Distinct(*Table*, *Formula*)

- *Table* - Required. Table to evaluate across.
- *Formula* - Required. Formula to evaluate for each record.

For this exercise we need to insert a drop down



In the **Items** property use the following formula: `Distinct(CitiesPopulation,Country).Result`



Now we need to change the Items formula for the gallery as follows.
`Filter(CitiesPopulation, Country = Dropdown1.Selected.Result)`



Using Search

In many apps, you can type one or more characters into a search box to filter a list of records in a large data set. As you type, the list shows only those records that match the search criteria.

Syntax

Search(*Table*, *SearchString*, *Column1* [, *Column2*, ...])

- *Table* - Required. Table to search.
- *SearchString* - Required. The string to search for. If *blank* or an empty string, all records are returned.
- *Column(s)* - Required. The names of columns within *Table* to search. Columns to search must contain text. Column names must be strings and enclosed in double quotes. However, the column names must be static and cannot be calculated with a formula. If *SearchString* is found within the data of any of these columns as a partial match, the full record will be returned.

For this exercise we need to insert a text input control

Now we need to change the Items formula for the gallery as follows.

`Search(CitiesPopulation,TextInput2.Text,"Country","City")`

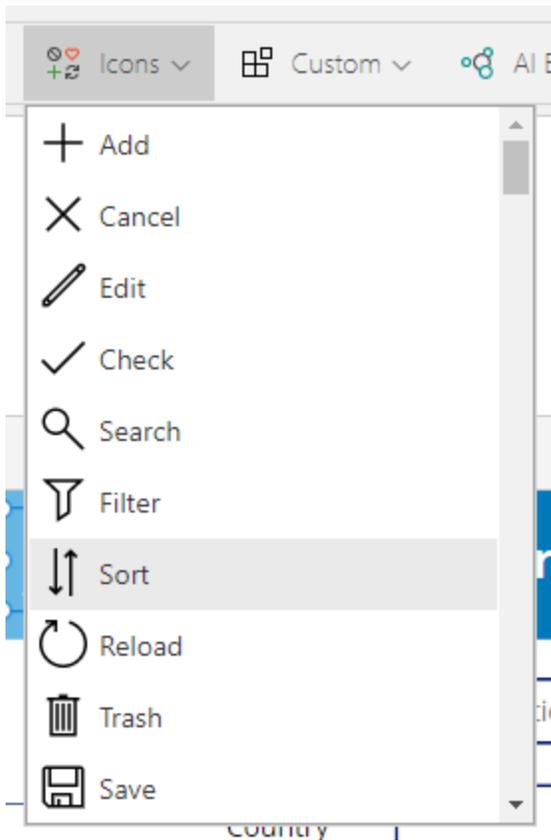


Using SortByColumns

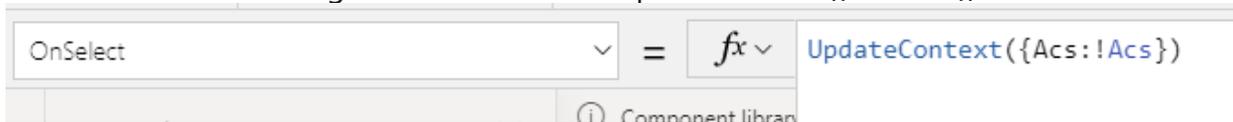
The **SortByColumns** function can also be used to sort a table based on one or more columns.

The parameter list for **SortByColumns** provides the names of the columns to sort by and the sort direction per column. Sorting is performed in the order of the parameters (sorted first by the first column, then the second, and so on). Column names are specified as strings, requiring double quotes if directly included in the parameter list. For example, **SortByColumns(CustomerTable, "LastName")**.

For this exercise we need to insert a sort icon



1. Make sure that the icon is still selected and the selected event for the icon is **OnSelect**.
2. Use the following formula **OnSelect** `UpdateContext({Acs:!Acs})`



Now we need to change the Items formula for the gallery as follows.

```
SortByColumns(
CitiesPopulation,
"Population",
If(
Acs,
Ascending,
Descending
)
)
```

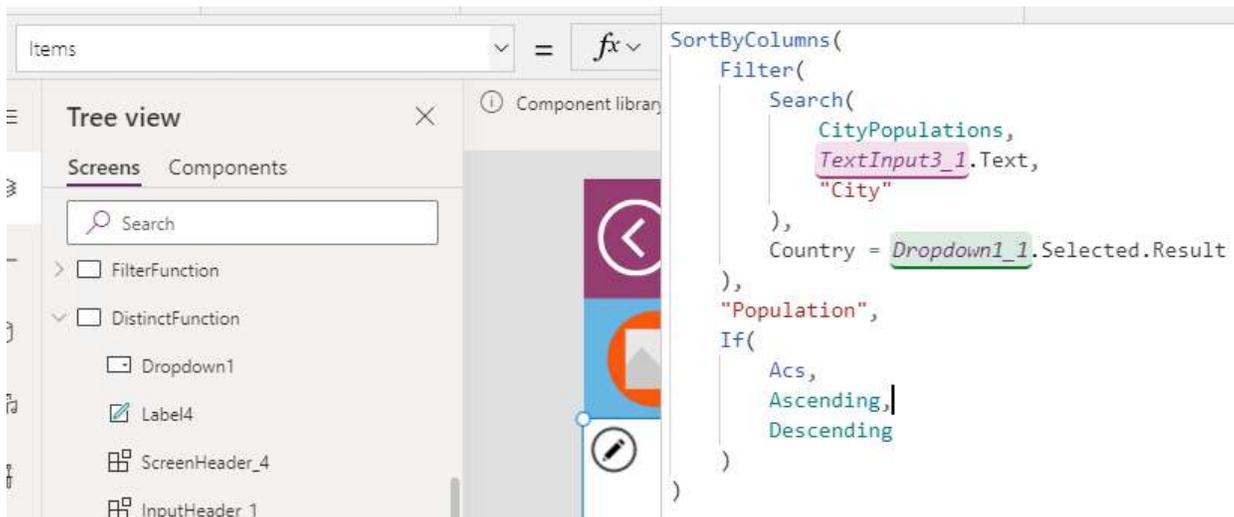
Remember that you can use a combination of Search, Filter and sort all together or independently

For example:

```

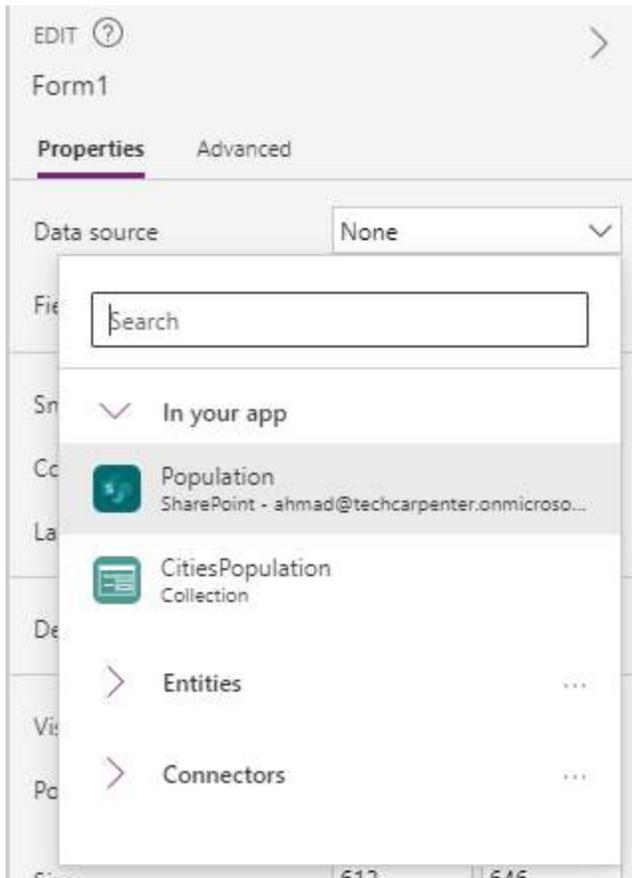
SortByColumns(
  Filter(
    Search(
      CityPopulations,
      TextInput3_1.Text,
      "City"
    ),
    Country = Dropdown1_1.Selected.Result
  ),
  "Population",
  If(
    Acs,
    Ascending,
    Descending
  )
)

```

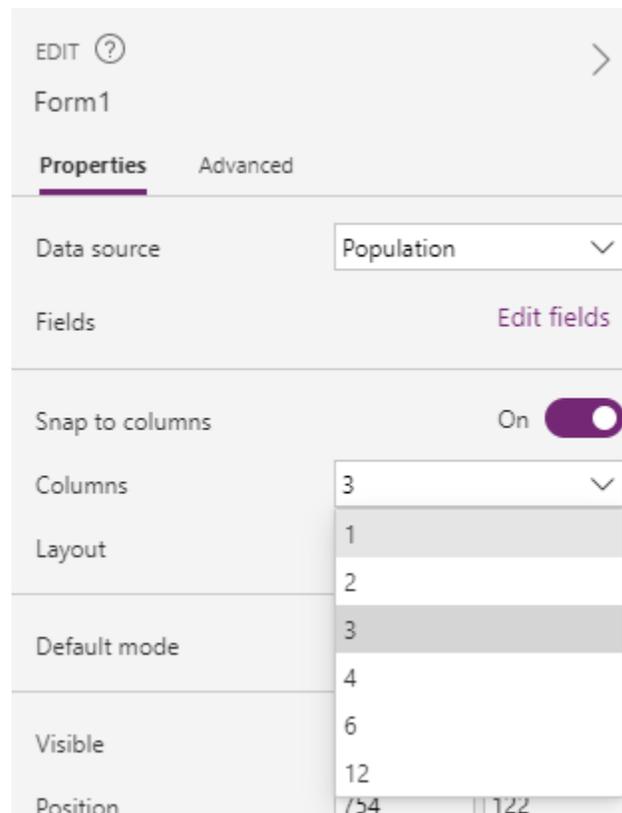


Adding new Items

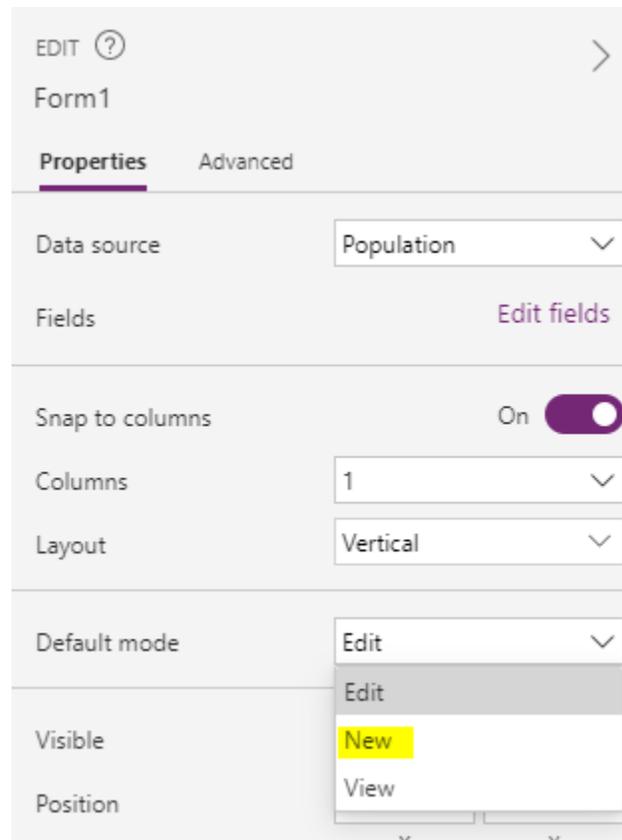
1. Add edit form from the insert tab menu on the top
2. Make sure that the form is still selected, from the properties pane on the right change the data source to the population SharePoint list.



3. From the same properties pane, change columns to 1



4. From the same properties pane, change the Default mode to New



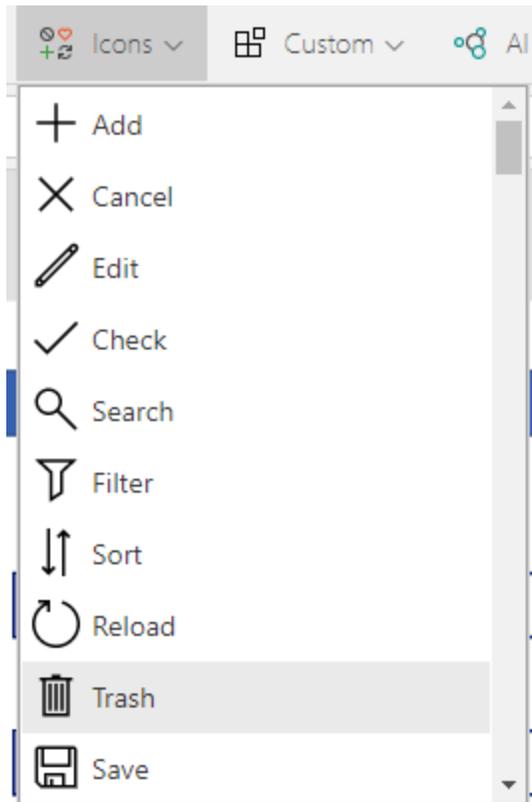
5. Add a new button and change the text on it to save

6. Make sure that the button is still selected, and paste the following code OnSelect event

```
SubmitForm(Form1);ResetForm(Form1);Refresh(Population);ClearCollect(CitiesPopulation,Population)
```

Remove Items

1. Select the first item in the gallery control (NOT at runtime)
2. Then from the insert tab add a Trash icon



3. The Icon should add on all rows of the gallery control



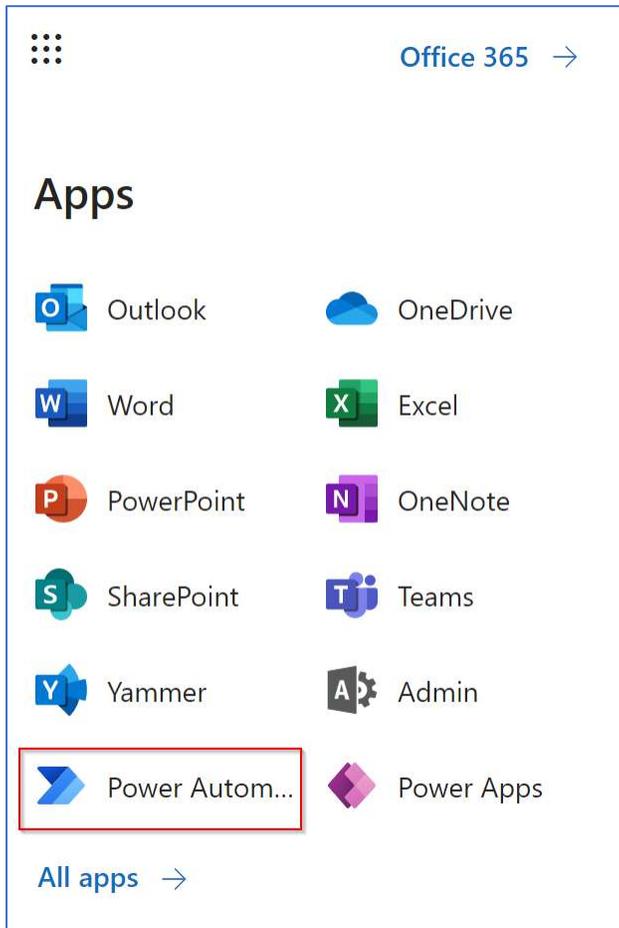
4. Make sure that the icon is still selected, and add the following OnSelect event for the icon:
Remove(Population,ThisItem);Refresh(Population);ClearCollect(CitiesPopulation,Population)

Congratulations! You just created your first App!

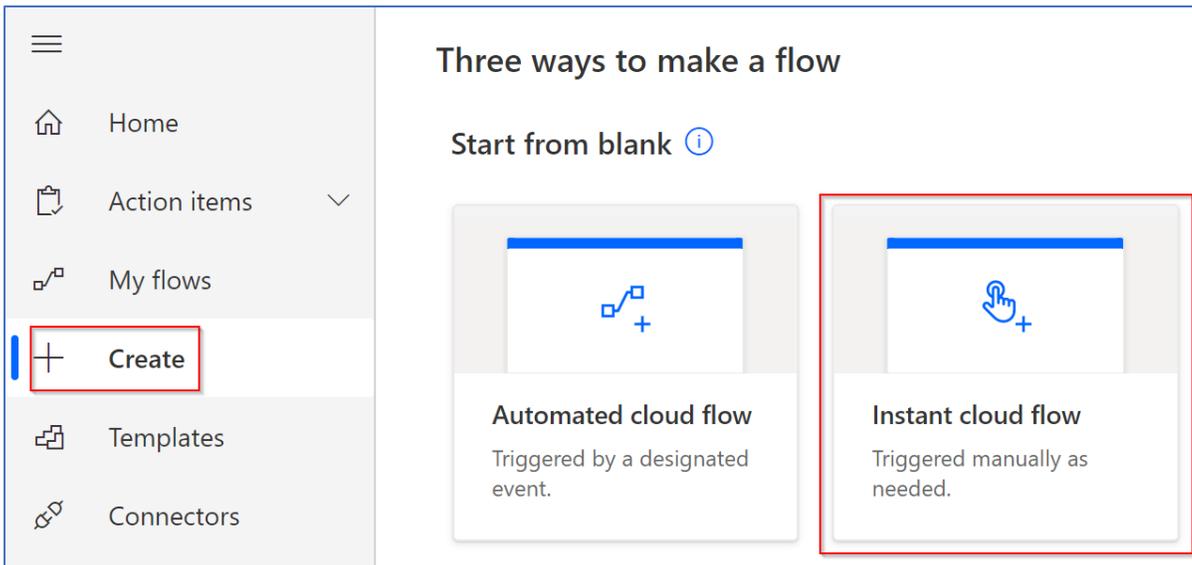
Lab 2 - Create an approval workflow triggered by the Power App

We're going to build a Power Automate flow which accepts data entered in the Power App, writes it to our SharePoint list (in an unapproved state), requests approval of the item, and then updates the Approval field if it's approved.

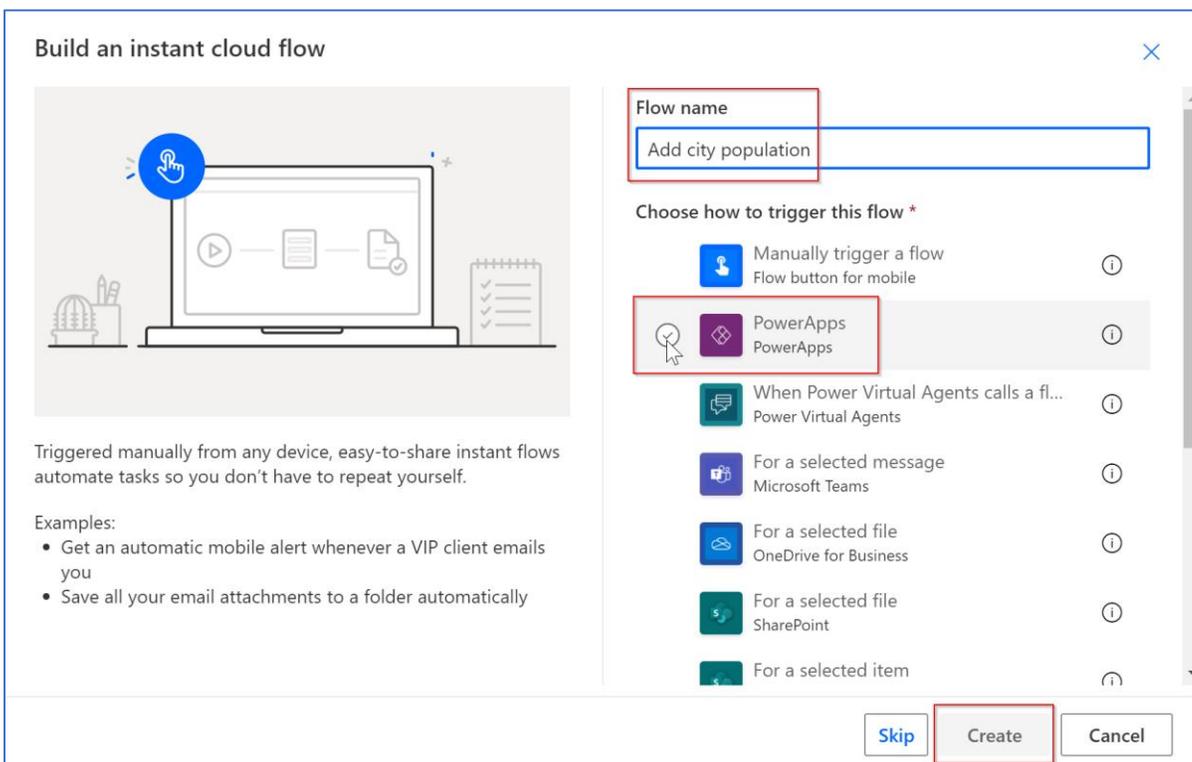
Open **Power Automate** from the App Launcher (Microsoft 365 "waffle" menu). You may need to click "All Apps" to find the Power Automate app.



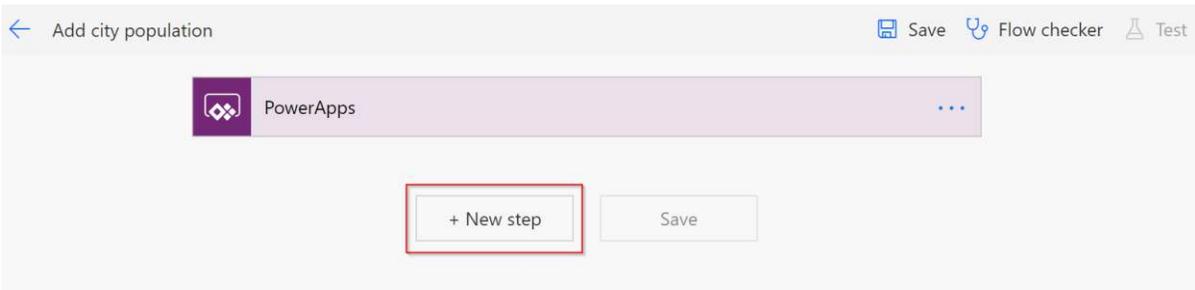
In the Power Automate maker portal, click **Create**. If this is your first time using Power Automate in this tenant, you'll be prompted to enter your country. We're going to create an **Instant cloud flow**, so click that tile.



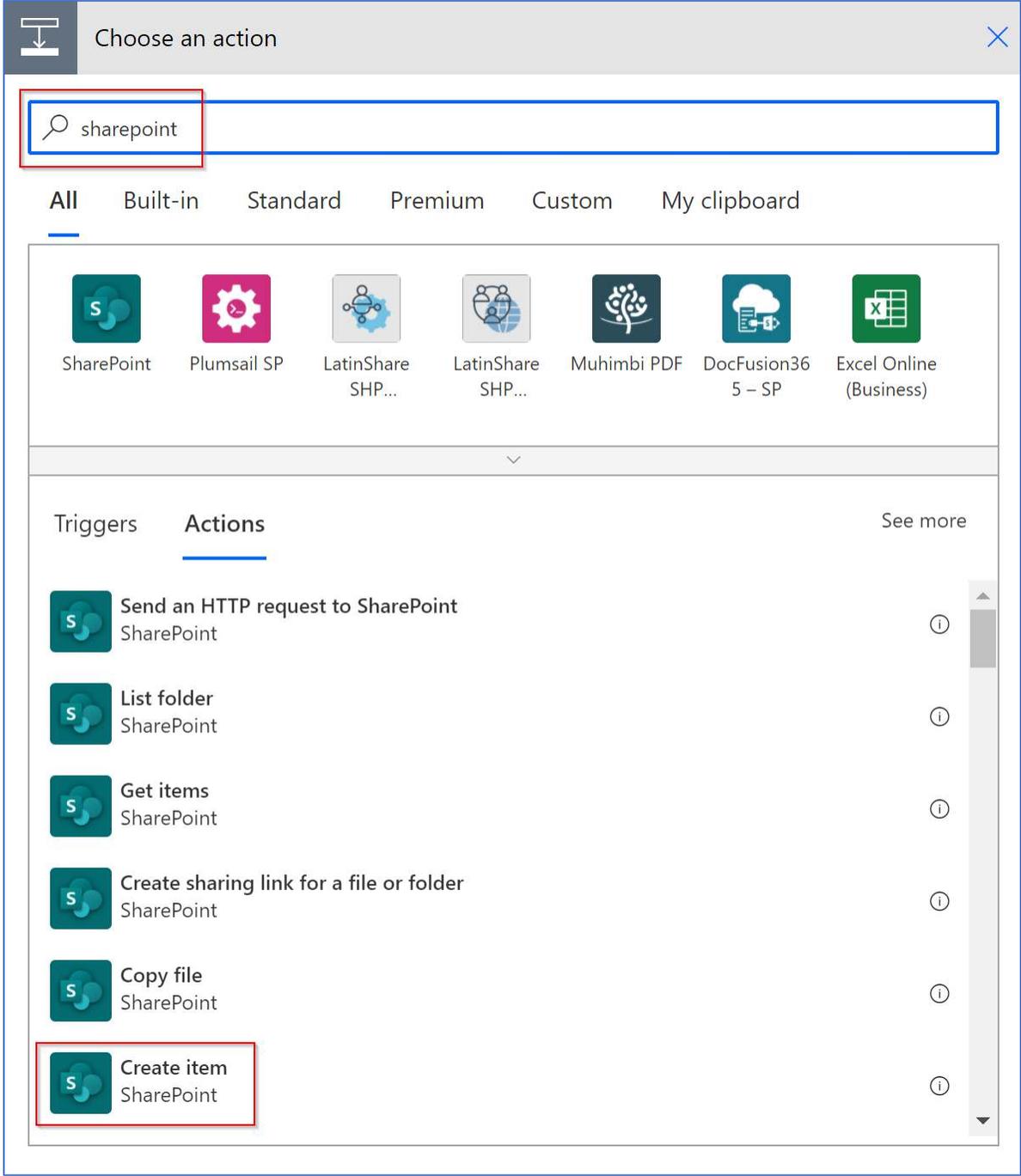
In the “Build an instant cloud flow” dialog, give your flow a name, such as **Add city population**. Select **PowerApps** as the trigger. Click **Create**.



You'll be taken to the flow design canvas, with a Power Apps trigger in place. Nothing needs to be entered in the trigger. Click **+ New step**.



The first step is to write the new data to the SharePoint list, but mark it as Not approved. *Search* for SharePoint actions, and then select **Create Item**.



In the *Site Address* field, select the site where you created the SharePoint list. In the *List Name* field, select your list. The fields from the list will then appear, as below.

The screenshot shows the 'Create item' dialog in SharePoint. The 'Site Address' field is set to 'Power Of The Cloud - https://m365x379358.sharepoint.com/sites/PowerOfTheCloud' and the 'List Name' field is set to 'City Populations'. Other fields include 'Title', 'Country', 'Population', and 'Approval Status' (set to 'No'). A 'Show advanced options' link is visible at the bottom left.

We want the flow to get the field values from the Power App, so for each field, we'll use the "Ask in Power Apps" dynamic data token. Here's how: Click into the *Title* field in the *Create Item* action (*Title* should be the column in your SharePoint list that contains the city name). You should see the *Dynamic content* dialog appear to the right. Click **Ask in PowerApps**.

The screenshot shows the 'Create item' dialog with the 'Dynamic content' dialog open. The 'Title' field is selected, and the 'Dynamic content' dialog shows the 'Ask in PowerApps' option under the 'PowerApps' category.

You should now see a purple dynamic data token in the *Title* field. Note that its name is a combination of the action name (Create item) and the field name (Title). This is what you'll see in Power Apps later. **TIP:** If you are doing this in an action with a longer name, you may want to rename the action before clicking *Ask in PowerApps*. This is also where you'll see it's a good idea to have short SharePoint internal column names, with no spaces.

* Site Address: Power Of The Cloud -
 https://m365x379358.sharepoint.com/sites/PowerOfTheCloud

* List Name: City Populations

Title: Createitem_Title x

Next, click into the *Country* field. In the *Dynamic content* dialog you'll likely see only the *Createitem_Title* token, but we want to *Ask in PowerApps* again, so click **See more**. You should then see another **Ask in PowerApps** – click it, to add *Createitem_Country* into our action.

Add dynamic content from the apps and connectors used in this flow. Hide

Dynamic content Expression

Search dynamic content

PowerApps See more

Createitem_Title

Do the same for the *Population* field, so your *Create item* action should now look like below (make sure Approval Status is set to **No**):

Create item

* Site Address: Power Of The Cloud -
 https://m365x379358.sharepoint.com/sites/PowerOfTheCloud

* List Name: City Populations

Title: Createitem_Title x

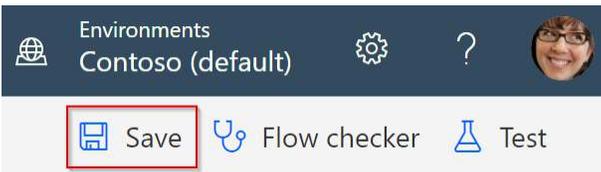
Country: Createitem_Co... x

Population: Createitem_Po... x

Approval Status: No

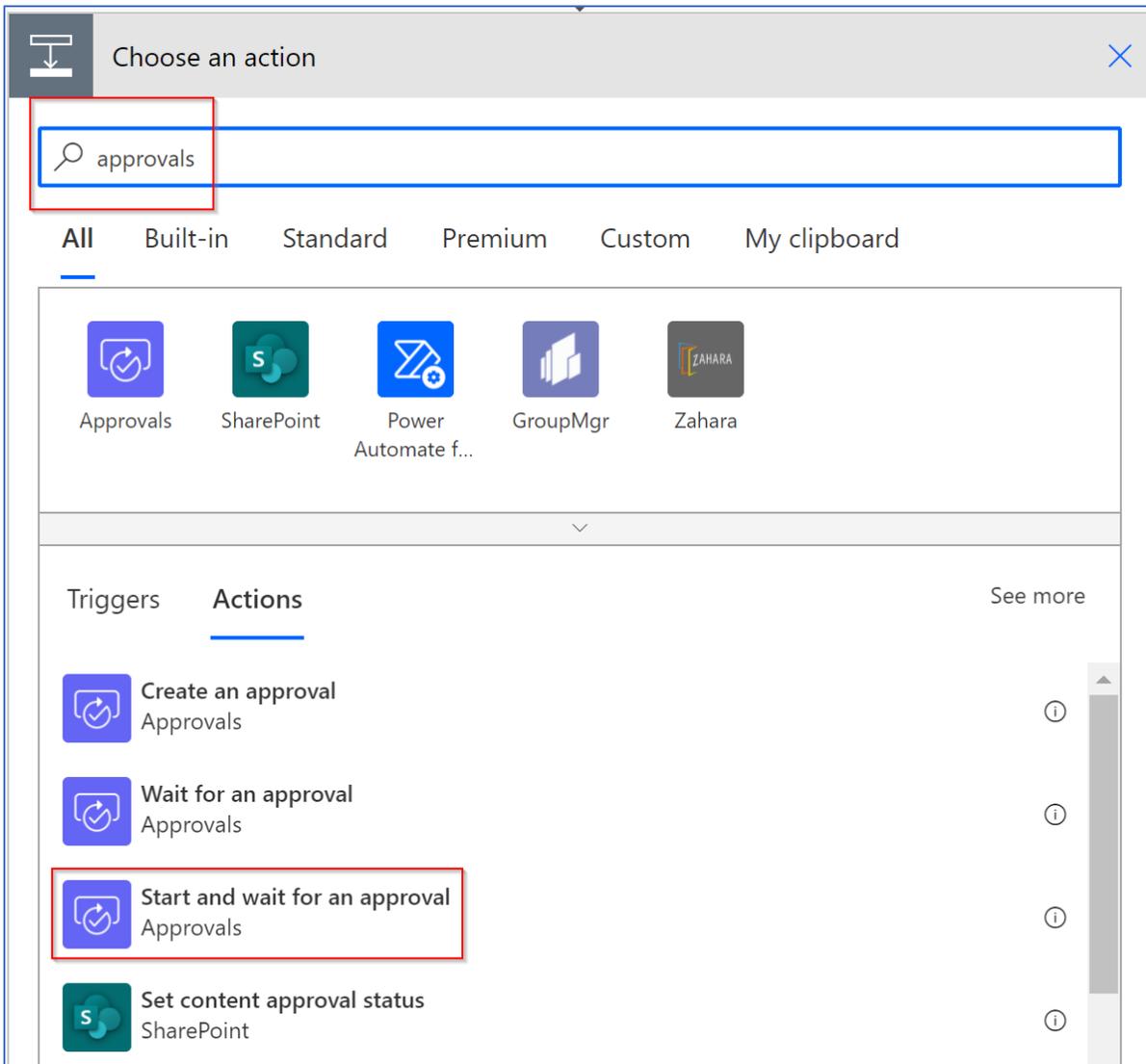
Show advanced options

At this point, we have a trigger and an action, so Power Automate allows us to Save the flow, which you should do. Click **Save**.



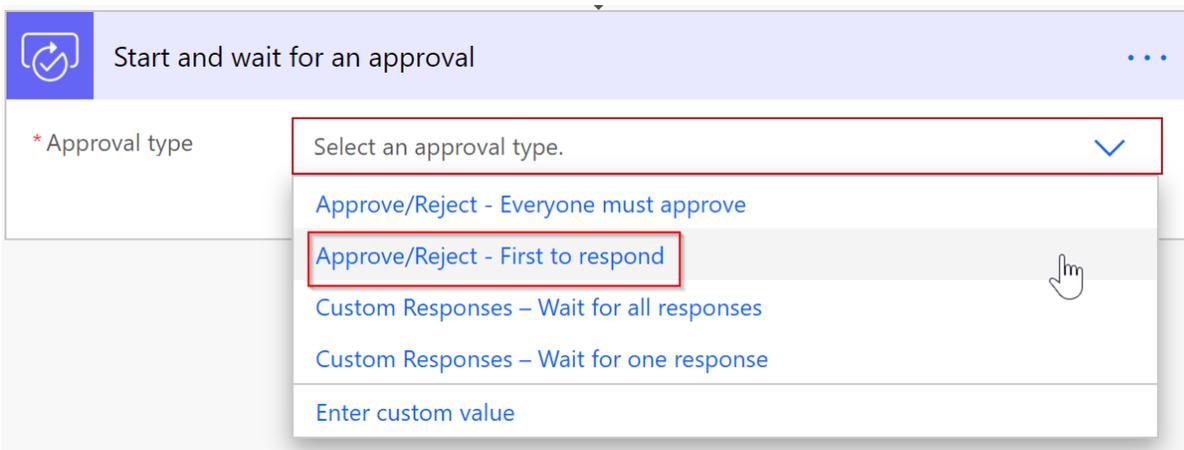
Next, we want to get this entry approved. NOTE: We could have added the approval process first, and not written the data to the SharePoint list at all until it was approved. But for demonstration purposes, we're saving items as unapproved, and then approving them.

To begin an approval process in Power Automate, click **+New Step** below the Create item action, and then search for **Approvals**. Select **Start and wait for an approval**.

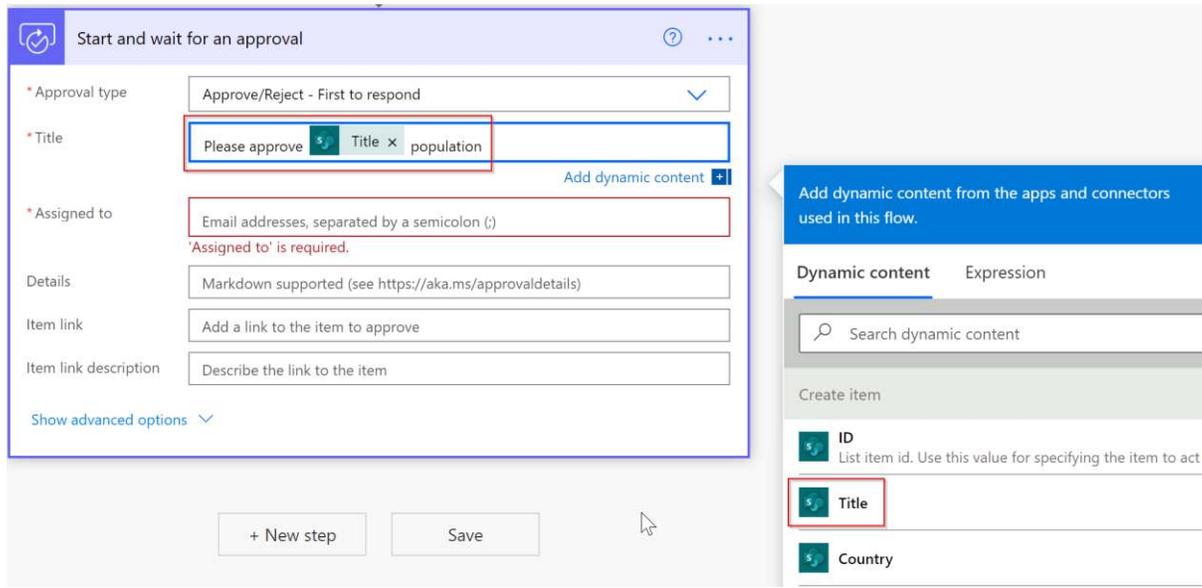


If this is the first time you've used an Approvals action in this tenant, you'll briefly see a *Create a connection* dialog, but it should automatically connect, as Approvals just uses your current login.

In the *Approval type* field, select **Approve/Reject – First to respond**, because we're only going to name one approver.



You'll now see more fields to fill in. The *Title* of the Approval is what the approver will see as an email subject and as the title of a card in the Approvals Center. We'll enter a combination of text and dynamic content. In the *Title* field, type **Please approve** and then select the **Title** token from the SharePoint *Create item* action, and then type the word **population**. Recall that the list's Title column is the name of the city. Thus, when this flow runs, the Approval title will read (for example, if New York were entered in the Power App): *Please approve New York population*



Next, we must assign the Approval to one or more approvers. Click into the *Assigned to* field. Note from the hint text that Power Automate requires an email address here. This could also be dynamic content – for example an email address that has been provided in a previous step, or perhaps the manager of the person who entered the data in Power Apps (obtained via an Office 365 Users *Get Manager* action). In our case, we're going to manually enter an individual account.

Begin typing the name of another account in your tenant that you are able to log in as for testing. If you do not have another account available, use your own. After you type some letters, Power Automate should suggest an account for you to click on.

Enter whatever you wish in the *Details* field (see an example below). You may wish to give the approver enough information here that they don't need to actually open the item to view it in the list. As you enter text and dynamic data tokens, you may find that the cursor jumps around, so pay attention to where it is before typing. Also, if you have trouble scrolling down in the *Dynamic content* dialog, you can begin typing the name of an element in the *Dynamic content* search box.

It's usually a good idea to provide a link to the item. In the *Item link* field, select **Link to item** from Dynamic content. Type whatever you like in the *Item link description* – this will present a friendly-looking link to the approver, rather than a long URL.

Your *Start and wait for an approval* action should now look something like this:

↻ Start and wait for an approval ? ...

* Approval type ▼
 Approve/Reject - First to respond

* Title
 Please approve s Title x population

* Assigned to
 MiriamG@M365x379358.OnMicrosoft.com;

Details
 New city population needs approval:
 City: s Title x
 Country: s Country x
 Population: s Population x

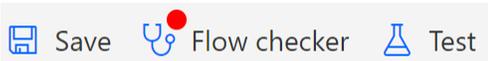
Item link
s Link to item x

Item link description
 Click to view the SharePoint list item

[Show advanced options](#) ▼

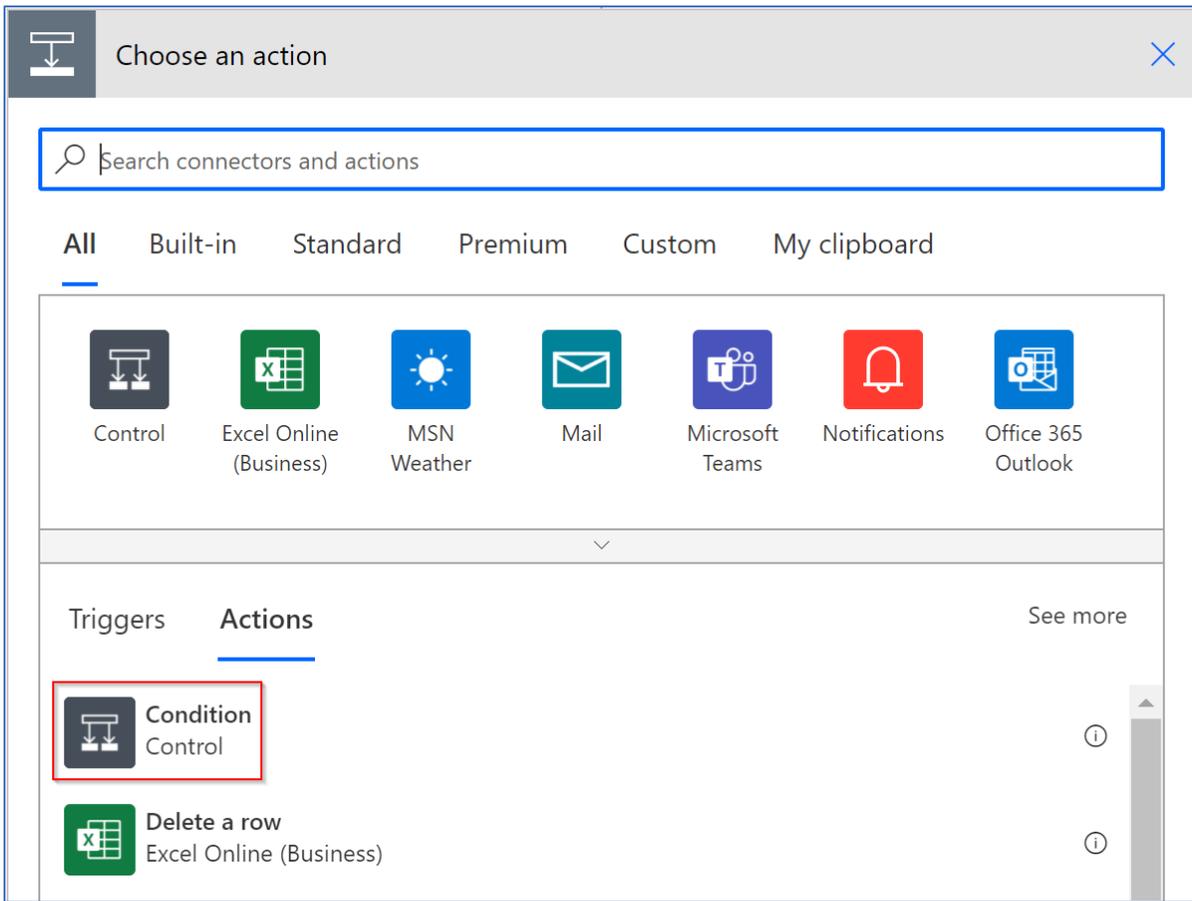
Save your flow again. It's a good idea to save after configuring any action that took some work 😊

If this is the first time you've used the Power Automate Approvals actions in your tenant, you'll likely notice an alert indicator in the *Flow checker*.

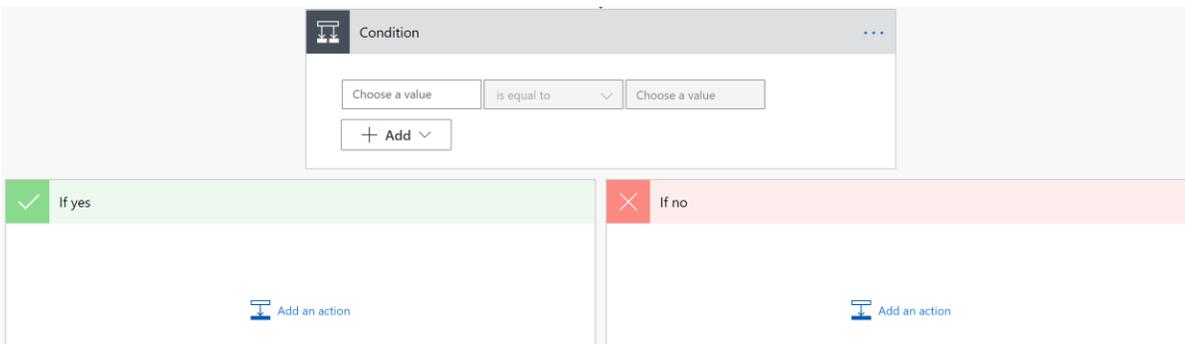


Clicking on that, you'll learn that a Common Data Service database does not yet exist in this environment. The Power Automate Approvals feature relies on Microsoft's CDS to function. As the message states, when you run this approval flow for the first time, a CDS database will automatically be provisioned. So we can ignore the message for now – it's just a warning, not an error.

At this point when it's running, the flow will stop and wait for the designated approver to respond (within 30 days), and then it will continue. So next the flow should do whatever you want it to do in either case (approve or reject). For this we need to add a Condition action, which is normally the first action listed in the *Choose an action* dialog.



Power Automate will add the *Condition* action, as well as *Yes* and *No* branches.



For an Approval, the Condition *value* we want to choose is the **Outcome** dynamic content token from the *Start and wait for an approval* action. The Outcome will be a single value derived from the possibly multiple values contained in the Responses (because you can have multiple approvers). We want to test whether **Outcome** (dynamic) *is equal to* **Approve** (type the word **Approve** in the *Choose a value* field).

Condition

Outcome x is equal to Approve

+ Add

If the approver has approved this entry, we want to change the Approval Status in the list to Yes. This can be done using the SharePoint *Update item* action in the *Yes* branch of the condition. Within the *Yes* branch, click **Add an action**.

✓ If yes

Add an action

Search for **SharePoint update** and select **Update item**.

In the *Update item* action, select the relevant site and list again, just as in the *Create item* action at the beginning.

Update item

* Site Address Power Of The Cloud -
https://m365x379358.sharepoint.com/sites/PowerOfTheCloud

* List Name City Populations

* Id Unique identifier of item to be updated
'Id' is required.

Title

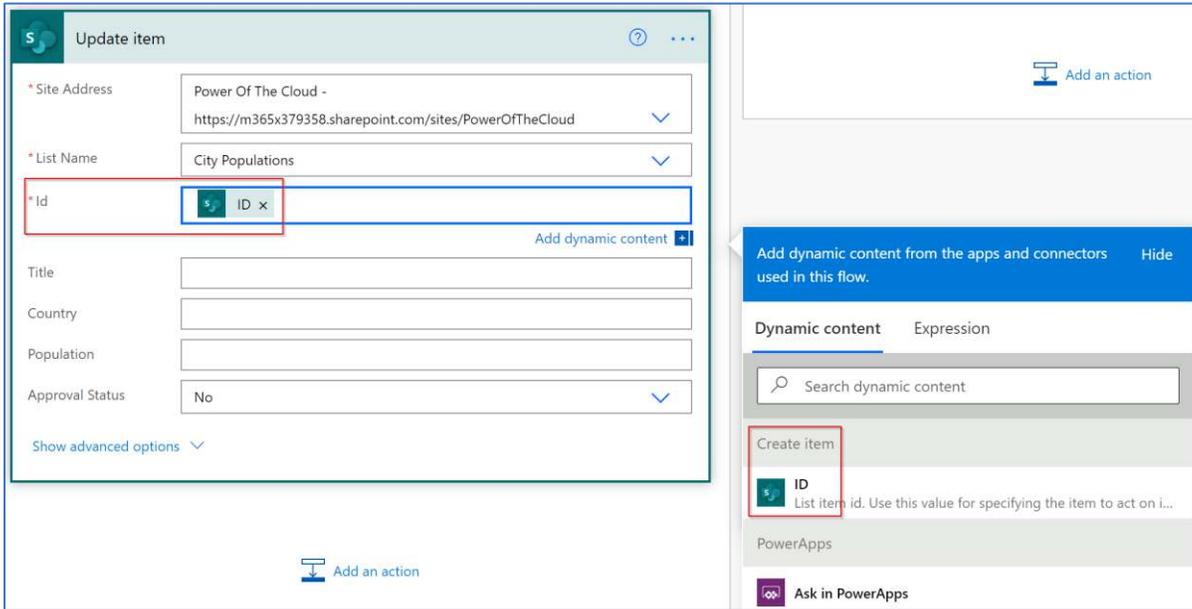
Country

Population

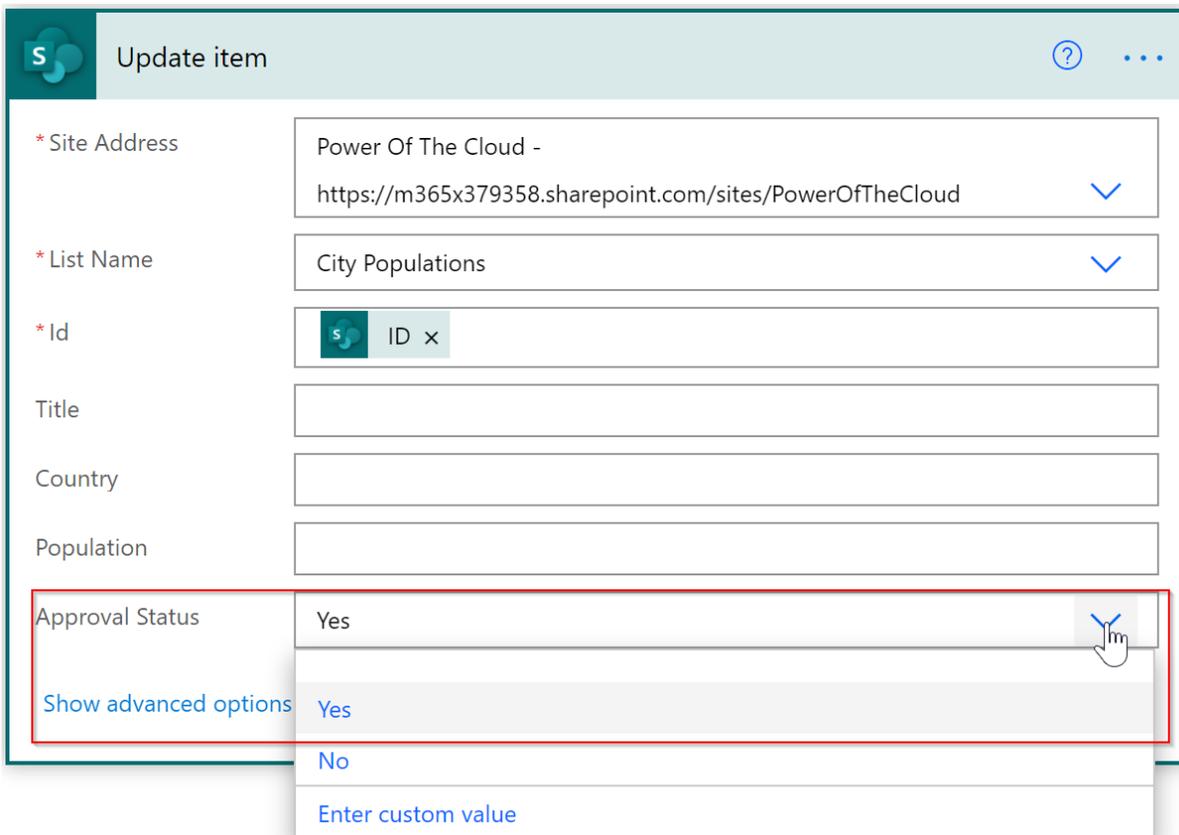
Approval Status No

Show advanced options

Next, the flow needs to know which item in that list to update. This will be the same item created in the previous *Create item* action, so that's the *Id* we must select from Dynamic content.



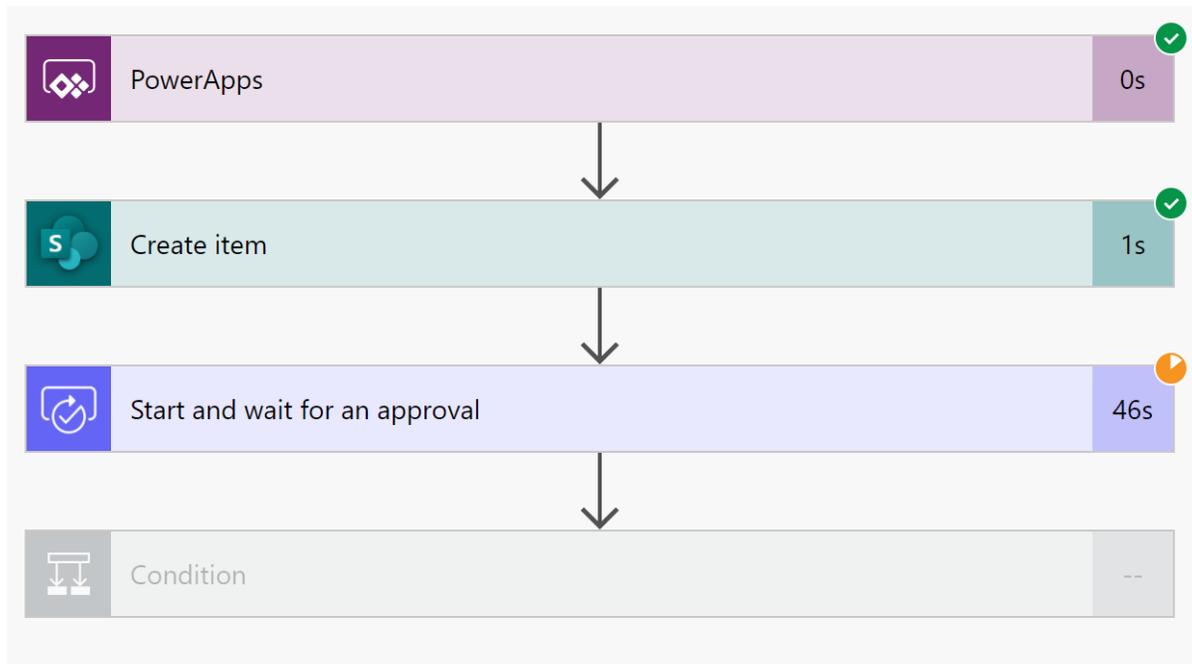
In the *Update item* action, you only need to enter values in the field/s you wish to update. Other values will remain the same. We only wish to change *Approval Status* to Yes, so select **Yes** instead of No.



In practice, you may also at this point wish to notify the person who added the item as to whether or not their entry was approved, but in the interest of time, we won't do that now.

IMPORTANT: If you have received the warning about there not yet being a CDS database for the Approvals connection, you'll need to run this flow once before attempting to run it from a Power App. To do this, we can use the Test feature, and enter the values manually that will be coming from the Power App. Click **Test flow** in the upper right. The default is to test Manually, so click the **Test** button at the bottom. Enter values for each field (you can make them up) and click **Run flow**. Your flow should start running. Click **Done** to close the test pane.

You should now see the flow's run, with green checkmarks on the first two steps, and then an orange circle on the Approval action, as it's waiting for input. This is as far as we need to go in order to provision the CDS database, so you can click **Cancel** in the upper right.



SAVE YOUR FLOW! Exit to this flow's dashboard by clicking the left arrow beside its name at the top of the canvas. From the dashboard page we can see when the flow runs.

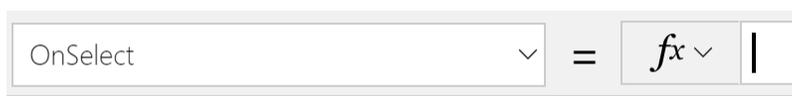


The next step is to connect this flow to your Power App.

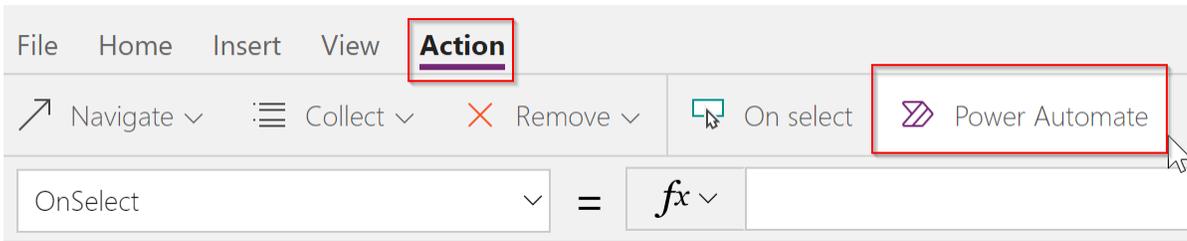
If you didn't already do so, add a **Button** control to the data entry screen of your app, and call it **Submit**. Make sure the button is on the screen itself rather than within some other control.



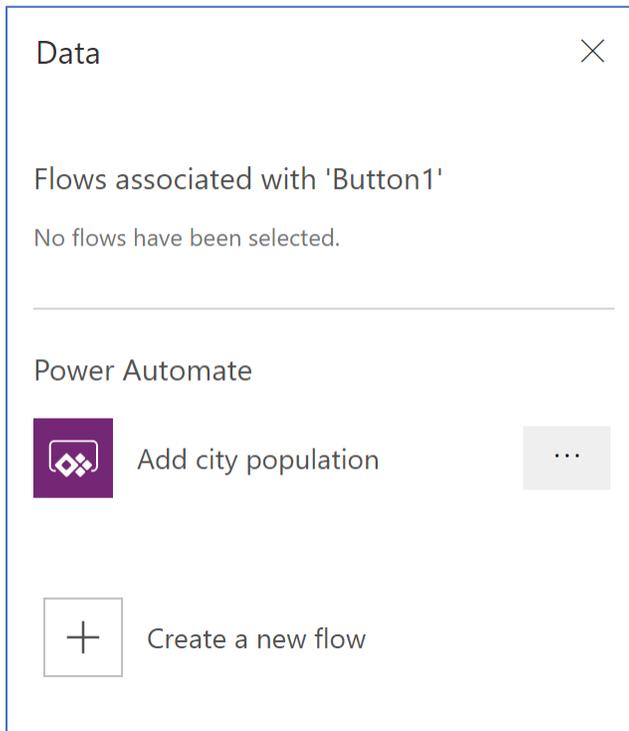
The *OnSelect* property should be the default property at the top of the canvas. Click into the formula bar and delete the word False which is the default.



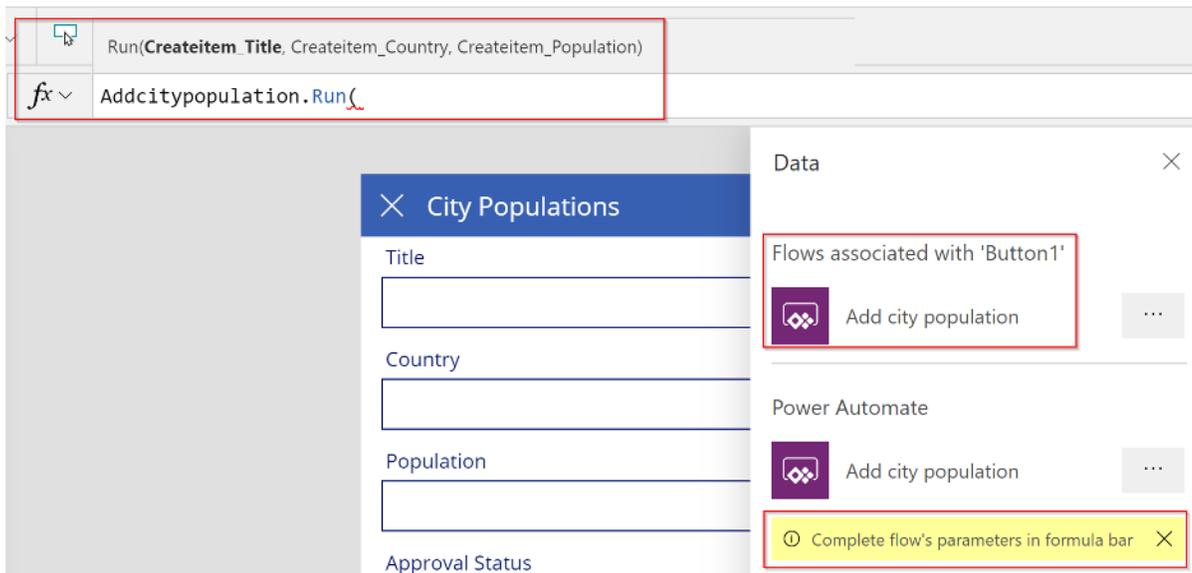
With your cursor still in the empty *OnSelect* formula bar, click on the **Action** menu, and select **Power Automate**.



A *Data* panel will open on the right, listing any flows in your environment which have Power Automate triggers. Select the flow we just created, and it will then be associated with your Button.



You'll see a message telling you to complete the flow's parameters in the formula bar, and a **Run** function (prefaced by the name of your flow) will have been started for you.



You can see from the tooltip that the function requires three parameters, which are the three Ask in Power Apps values we created within the flow – separated by commas. After the left parenthesis, use what you learned in the Power Apps workshop to enter the values which should be passed to the flow, in the correct order, separated by commas. Close the parentheses.

Your final formula will differ from the screenshot below, depending on what you called your Power App controls. In this example, the values are coming from Data Cards, but yours might be TextInput controls, in which case you would need to add .Text or .Value for the number.



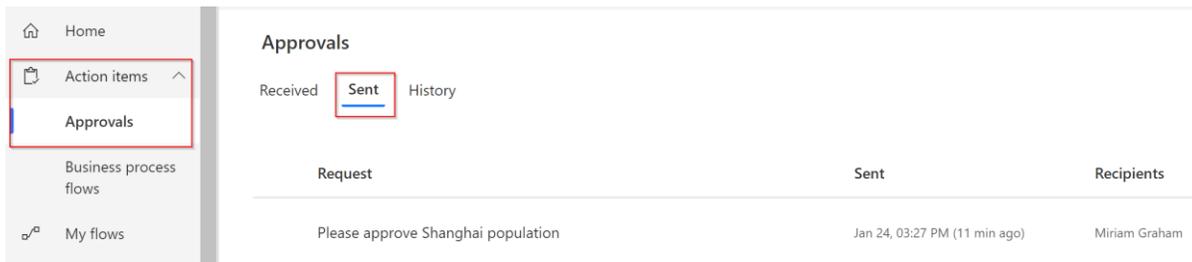
Save your Power App. You may wish to add a version note to the effect that you added a Submit button which triggers a Power Automate flow.

To test, play the Power App, and enter City, Country, and Population (you can either make it up or look it up). Click the new **Submit** button.

Switch over to the flow’s dashboard, and you should see that it’s running, having been triggered by the Power App’s Submit button.

28-day run history ⓘ			↻ All runs
Start	Duration	Status	
Jan 24, 03:27 PM (9 min ago)	00:09:19	Running	
Jan 24, 03:18 PM (18 min ago)	00:18:12	Test canceled	
Jan 24, 03:06 PM (30 min ago)	06 ms	Succeeded	

Remember that the flow will pause to wait for the approval, so it will stay in a *Running* Status until the approver takes action. You can view the sent approval in the Power Automate Approvals center.



While the flow is in this paused state, take a look at your City Population list in SharePoint. You should see your new entry, but the Approval Status is No.

If you specified yourself as the approver, you should also see this approval in the *Received* tab in the Approvals Center. Or you can check the email of the account which you specified as the *Assigned to* approver.

The approval request in the Approvals Center looks like this:

Respond ✕

Overview

Approval
Please approve Shanghai population

Requester
 Megan Bowen

Received
Jan 24, 04:43 PM (32 sec ago)

Link
[Click to view the SharePoint list item](#)

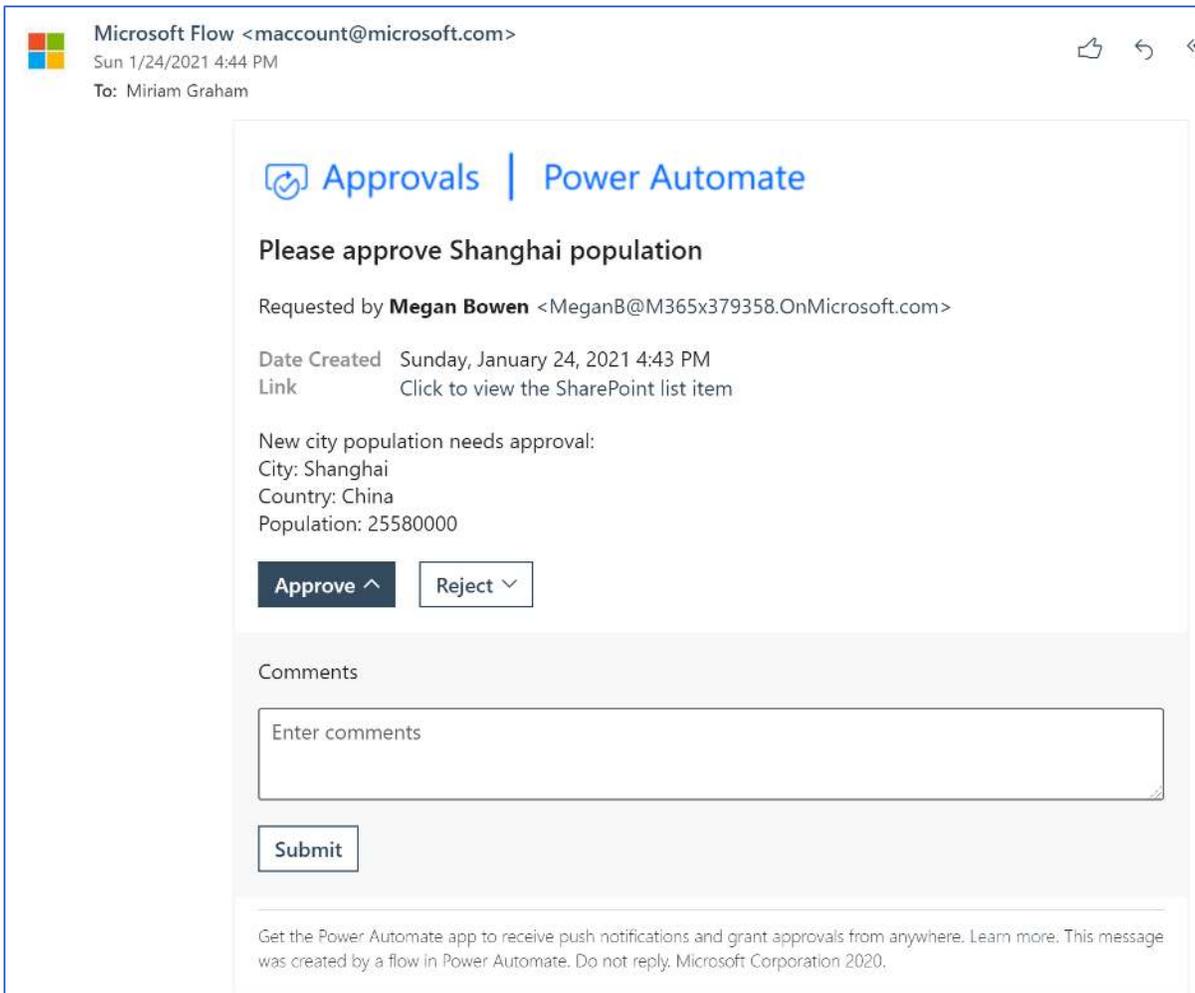
Details

New city population needs approval:
City: Shanghai
Country: China
Population: 25580000

Choose your response 

- Approve
- Reject
- Reassign

The approval email looks like this:



Whichever way you get there, click **Approve** to approve the item. If this is the first time this approver has used the Power Automate approvals, then even if they click Approve in the email, they'll be taken to the Power Automate Approvals Center to create their Approvals connection.

Going back to the flow dashboard, you should shortly see the flow run marked as *Succeeded*. And going back to your SharePoint list, you should see that the Approval Status has changed to Yes.

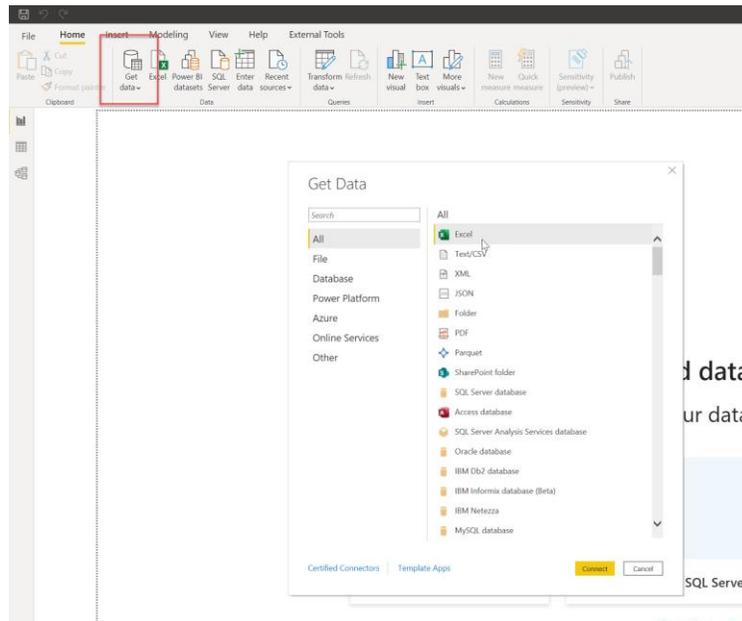
Congratulations! You have now completed your Power Automate Lab!

Lab 3 - Building the Population Report – Data Prep

Within this section of the lab we are going to create a new Power BI report that connects to our data in SharePoint. This report will display the cities and their populations. We will publish it to the Power BI portal, configure it to refresh its data, and finally we will embed it on the home page of our SharePoint site. In order to connect to data in SharePoint, you need to use Power BI Desktop. Power BI Desktop is available from the Windows Store, but can also be downloaded direct from <https://www.microsoft.com/en-us/download/details.aspx?id=45331> for those not using Windows 10. There are distinct advantages to using the store version such as automatic updating, and the fact that it can be installed without administrative rights. Ensure that Power BI Desktop is installed before proceeding.

Creating the Approval Report

1. Launch Power BI Desktop
2. Login using your own or the provided tenant credentials
3. Select Get Data from either the splash screen or the ribbon



4. Enter SharePoint into the search box, select SharePoint Online List, and click the Connect button
5. Enter the URL of your SharePoint site (not list) in the next dialog box, then click the OK button (use MS Authentication and sign in with the creds used previously)

SharePoint Online Lists

Site URL ⓘ

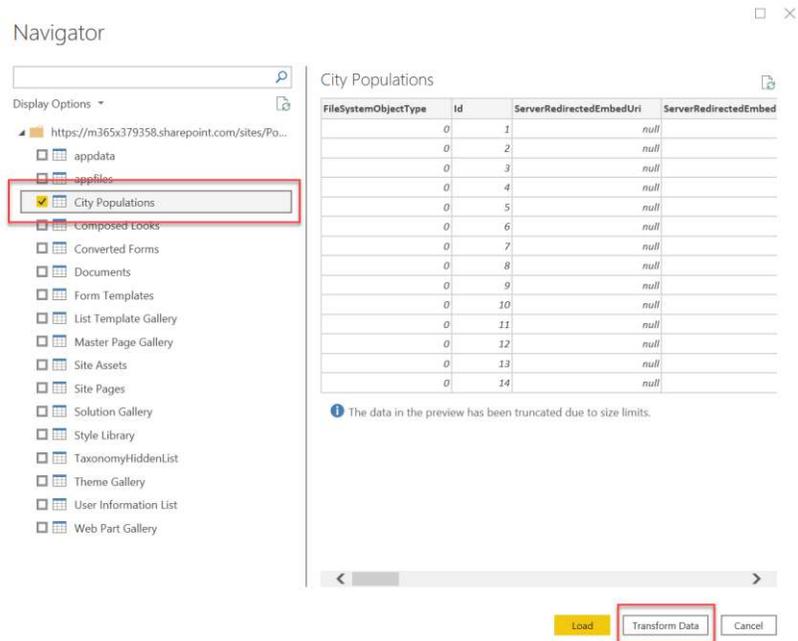
Implementation

- 2.0 (Beta)
- 1.0

▸ Advanced options

6. If you see the implementation option choose 1.0

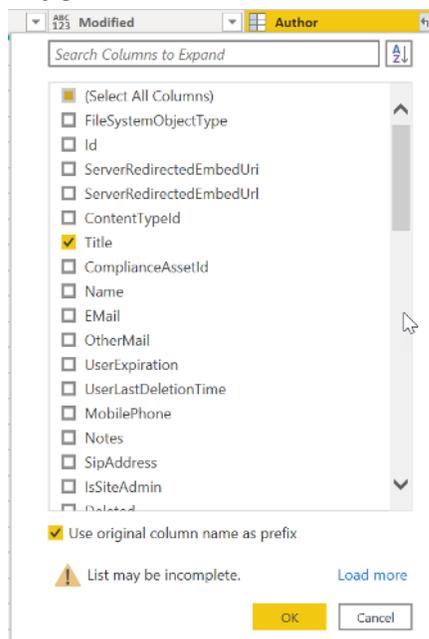
7. You will be presented with all of the lists available in the SharePoint site (including otherwise hidden lists). Select the City Populations list that you created earlier. In order to control the quantity and the shape of the date, click on the Transform Data button.



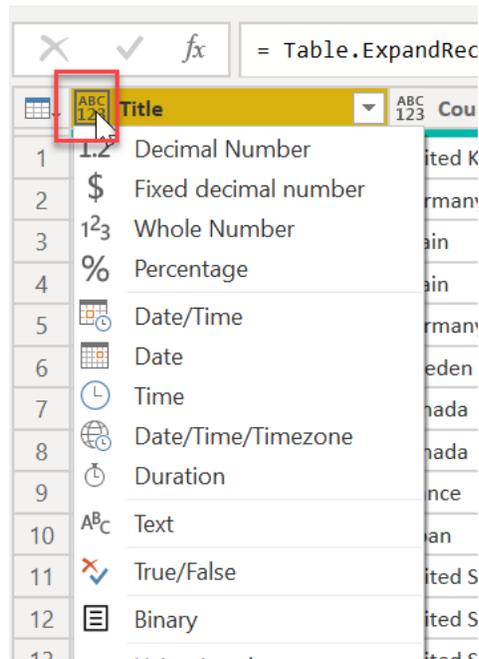
8. You are now in the Power Query editor. From here you can filter and transform the data for inclusion in the data model of the report. To begin, remove all of the columns that are unnecessary in the report. You remove a column by right clicking on its header and selecting Remove. Multiple columns can be selected for removal by ctrl-clicking on the column header before removing. Remove all of the columns from the table except for the following: Title, Country, Population, ApprovalStatus, Modified, and Author. When complete, your window should appear similar to the image below:

	ABC 123 Title	ABC 123 Country	ABC 123 Population	ABC 123 ApprovalStatus	ABC 123 Modified	Author
1	London	United Kingdom	8615000	TRUE	1/23/2021 11:43:27 PM	Record
2	Berlin	Germany	3562000	TRUE	1/23/2021 11:43:28 PM	Record
3	Madrid	Spain	3165000	TRUE	1/23/2021 11:43:31 PM	Record
4	Barcelona	Spain	1620000	TRUE	1/23/2021 11:43:38 PM	Record
5	Frankfurt	Germany	753000	TRUE	1/23/2021 11:43:45 PM	Record
6	Stockholm	Sweden	2400000	TRUE	1/23/2021 11:44:27 PM	Record
7	Toronto	Canada	2930000	TRUE	1/23/2021 11:43:49 PM	Record
8	Vancouver	Canada	675000	TRUE	1/23/2021 11:43:54 PM	Record
9	Paris	France	2161000	TRUE	1/23/2021 11:43:55 PM	Record
10	Tokyo	Japan	9273000	TRUE	1/23/2021 11:43:55 PM	Record
11	Chicago	United States	2710000	TRUE	1/23/2021 11:43:55 PM	Record
12	New York	United States	8419000	TRUE	1/23/2021 11:43:55 PM	Record
13	Los Angeles	United States	3967000	TRUE	1/23/2021 11:43:55 PM	Record
14	Pittsburgh	United States	302000	TRUE	1/23/2021 11:44:03 PM	Record
15	Amman	Jordan	4000000	TRUE	1/23/2021 11:44:03 PM	Record
16	Cairo	Egypt	9000000	TRUE	1/23/2021 11:44:03 PM	Record
17	Rome	Italy	2873000	TRUE	1/23/2021 11:44:03 PM	Record
18	Milan	Italy	1352000	TRUE	1/23/2021 11:44:03 PM	Record
19	Moscow	Russia	12538000	TRUE	1/23/2021 11:44:03 PM	Record
20	Istanbul	Turkey	15190000	TRUE	1/23/2021 11:44:03 PM	Record

9. Notice that the Author column contains a series of Records. Each of these records is a lookup into another list, in this case, the hidden UserInformation list. We want to retrieve the name of the author. To do that, click on the small expand button on the right of the column header for **Author**. Then, deselect all the columns except for **Title**.



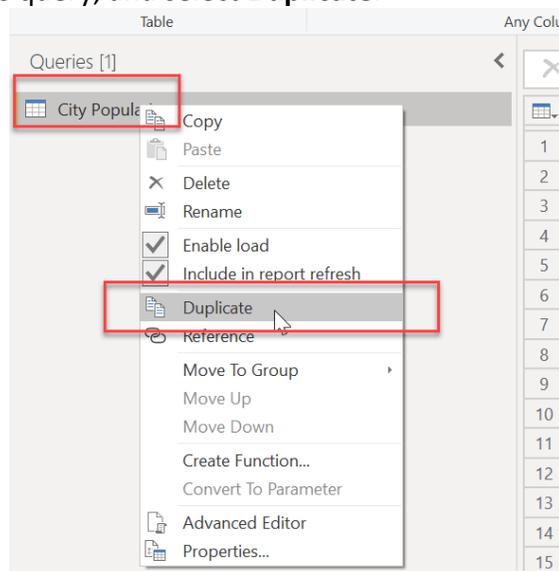
10. Change the name of the **Author.Title** column to simply **Author**. You can do this by double clicking on the column header.
11. Change the data types of the columns to reflect their true type. You can do this by selecting the type selector to the left of each column header and selecting the appropriate type.



12. Set the columns to the following types:

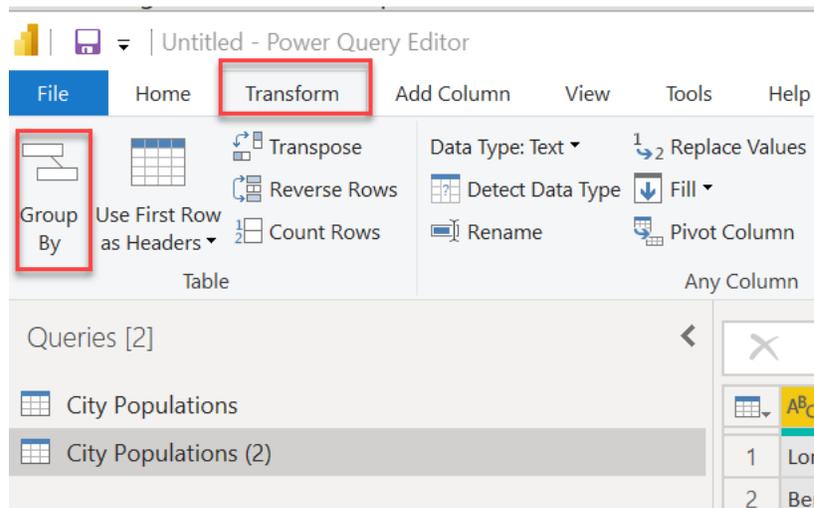
1. **Title** – Text
2. **Country** – Text
3. **Population** – Whole Number
4. **ApprovalStatus** – True/False
5. **Modified** – Date/Time
6. **Author** – Text

13. Notice that some of the city data is duplicated. We only want the most recent values for each city. To do this, we will first need to create a table with the cities, and their most recent update dates. Right click on the **City Populations** query, and select **Duplicate**.

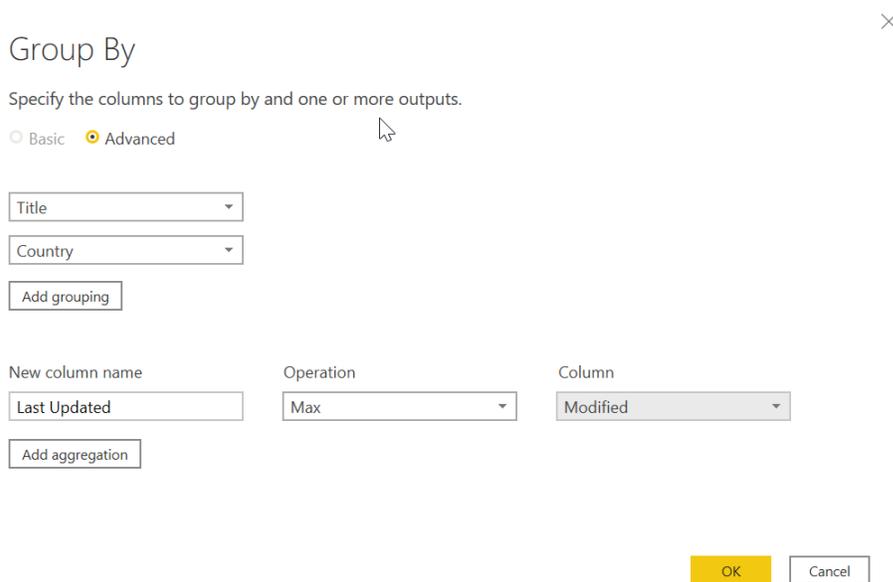


14. On the new query, remove all of the columns except for **Title**, **Country**, and **Modified**

15. From the **Transform** tab in the ribbon, select the **Group By** button.



16. Select the Advanced option to group by two columns. Select **Title** and **Country**. Enter **Last Updated** as the new column name, **Max** as the Operation and **Modified** as the Column. What this does is select unique values for **Title** and **Country**. When there are multiple rows with the same values, it will select the row with the Maximum value for **Modified**, and use that value as **Last Updated**.



17. Next, merge this table with the original table. From the Home tab, select the Merge Queries button. Select all three columns for the first query (by holding down the shift key when selecting). Select the original query as the query to be merged, and then select the **Title**, **Country**, and **Modified** columns. This will perform a Left Outer Join, which will retrieve the most recent details from the original table. When ready, select the OK button.

Merge

Select a table and matching columns to create a merged table.

City Populations (2)

Title	Country	Last Updated
London	United Kingdom	1/23/2021 11:43:27 PM
Berlin	Germany	1/23/2021 11:43:28 PM
Madrid	Spain	1/23/2021 11:43:31 PM
Barcelona	Spain	1/23/2021 11:43:38 PM
Frankfurt	Germany	1/23/2021 11:43:45 PM

City Populations

Title	Country	Population	ApprovalStatus	Modified	Author.Title
London	United Kingdom	8615000	TRUE	1/23/2021 11:43:27 PM	Megan Bowen
Berlin	Germany	3562000	TRUE	1/23/2021 11:43:28 PM	Megan Bowen
Madrid	Spain	3165000	TRUE	1/23/2021 11:43:31 PM	Megan Bowen
Barcelona	Spain	1620000	TRUE	1/23/2021 11:43:38 PM	Megan Bowen
Frankfurt	Germany	753000	TRUE	1/23/2021 11:43:45 PM	Megan Bowen

Join Kind

Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

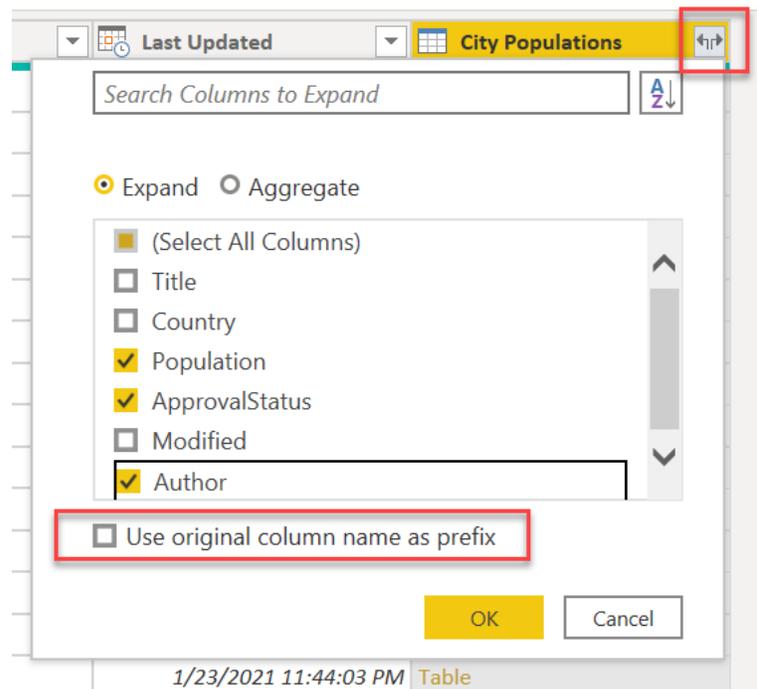
Fuzzy matching options

The selection matches 26 of 26 rows from the first table.

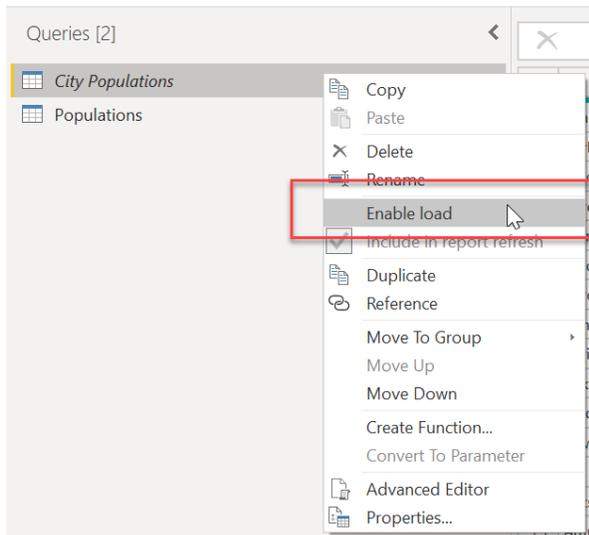
OK

Cancel

- Click the Expand icon in the right of the City Populations column and deselect all columns except for **Population**, **ApprovalStatus**, and **Author**. Also deselect **Use original column name as prefix**. Click OK when ready.



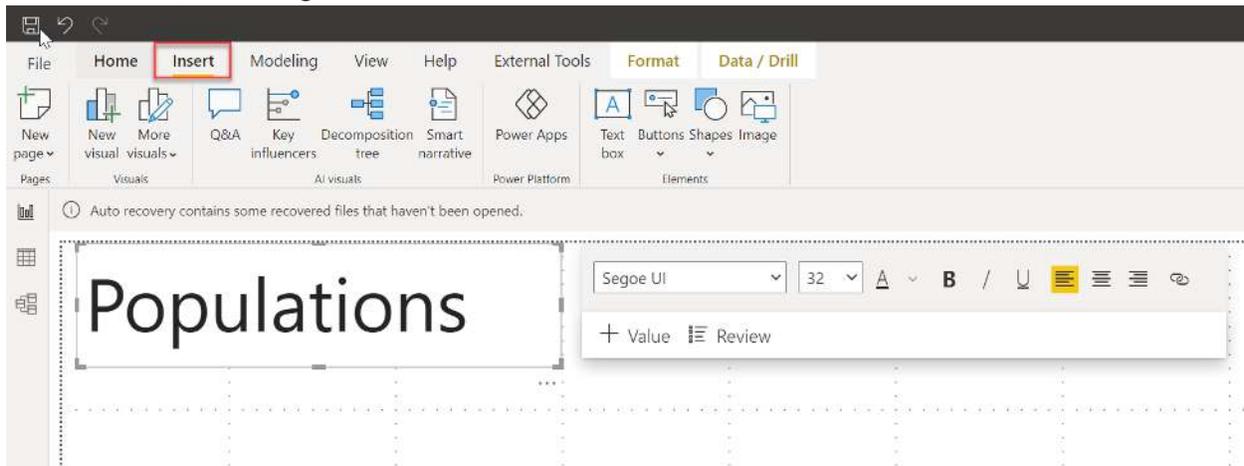
- Rename the Query to **Populations**. This can be done by double clicking on the query name in the query list, or in the **Query Settings** pane.
- It is not necessary to load both queries into the model, right click on the original query (**City Populations**) and deselect the Enable Load option.



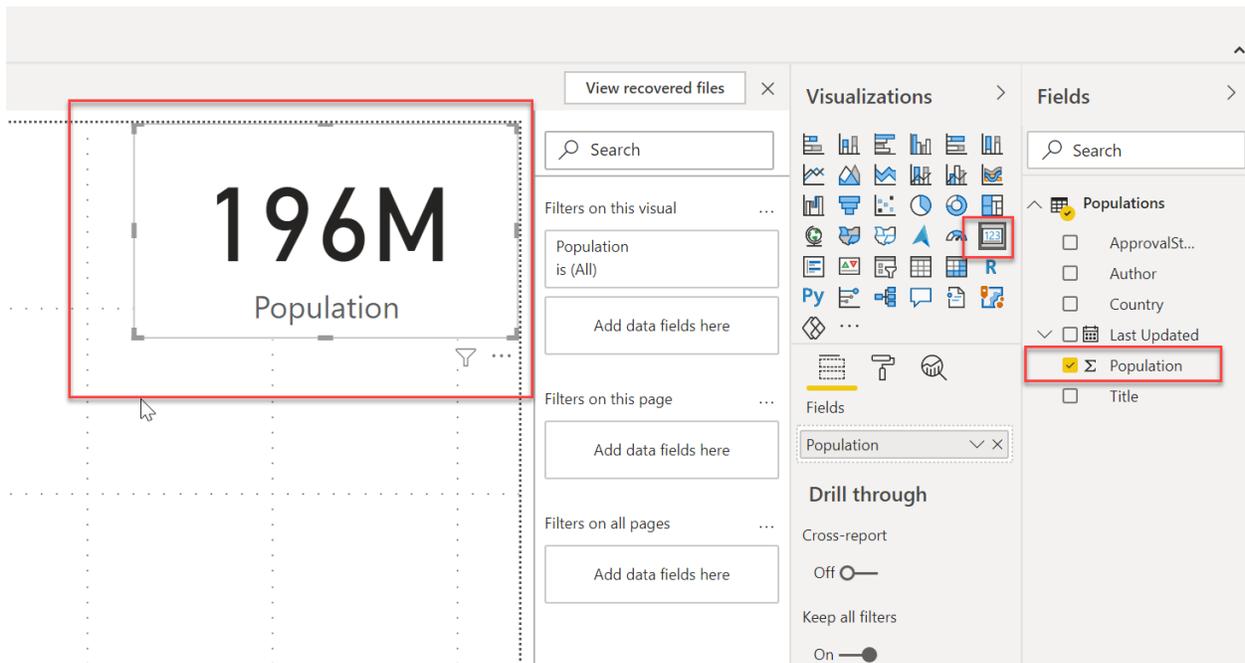
21. Select the **Close and Apply** button from the Home tab on the ribbon to close the Power Query editor, and start adding report visuals.

Building the Population Report – Visuals

1. Create a title for the report by selecting the **Text box** button from the **Insert** tab on the ribbon. Set the Font and size as desired, and change the size of the text box to match the title.



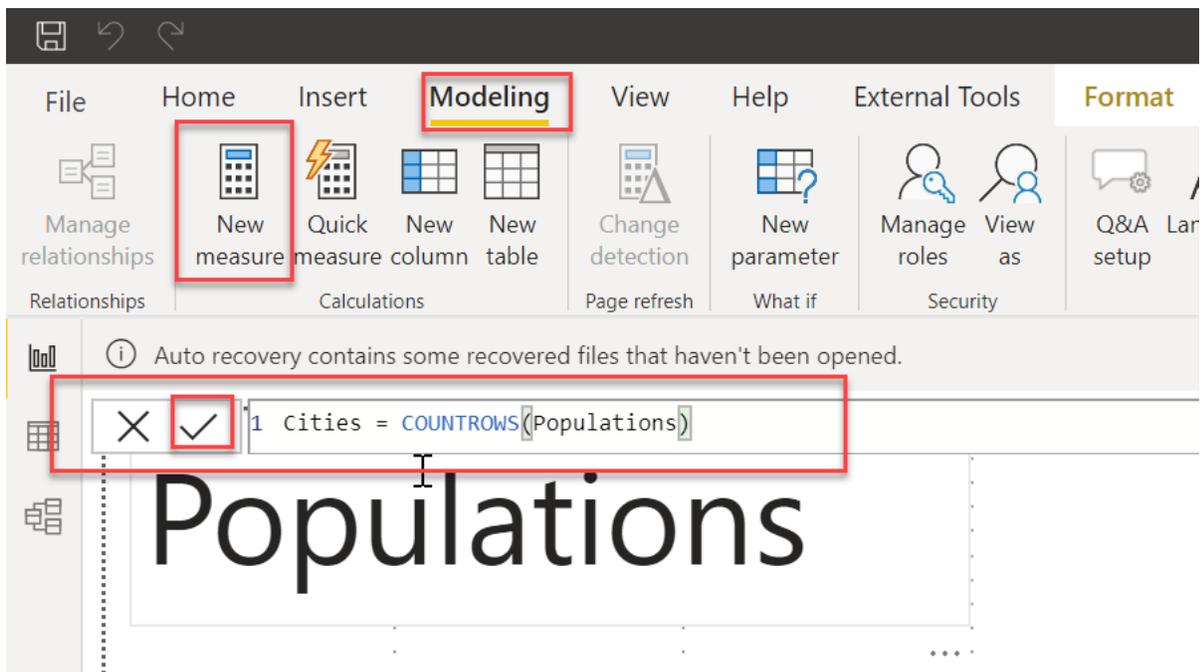
2. Add a card to display the population by clicking an empty space on the design canvas, and selecting **Population** from the Fields Pane. Power BI will create a column chart using **Population**. Change the visual to a Card by selecting the Card icon from the Visualizations pane. Drag the card to the upper right corner of the design surface and resize it appropriately. Using the **Population** field in this way creates an *implicit* measure, because it is a numeric field, and Sum is the default aggregation. It works in this case, but it is considered to be a best practice to use *explicit* measures, as you will do in the next step.



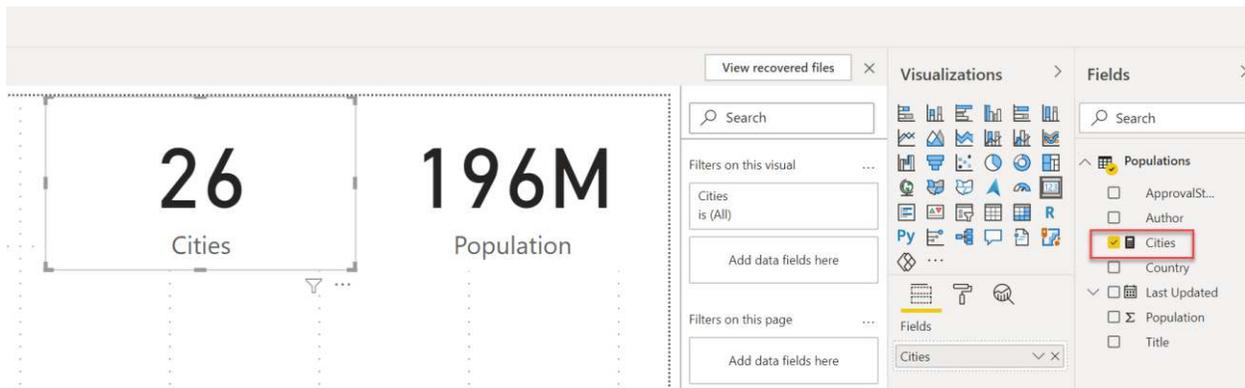
3. Create an explicit measure to show the number of cities. Click the **New Measure** button from the **Modeling** tab on the ribbon. In the formula editor, replace “Measure = ” with the following DAX expression:

Cities = COUNTROWS(Populations)

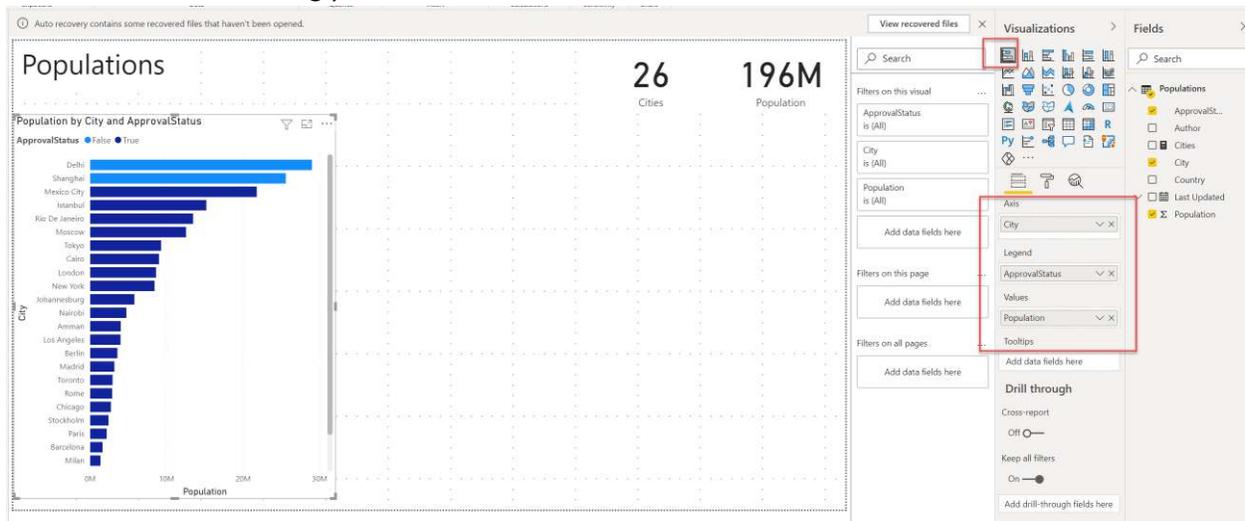
Given the fact that each city name is unique in the table, a count of the table rows will give the number of cities. Click the check box when done.



4. Add a card to the report and place it to the left of the **Population** card. Use the same technique used in step 2 above, replacing **Population** with your new **Cities** measure.



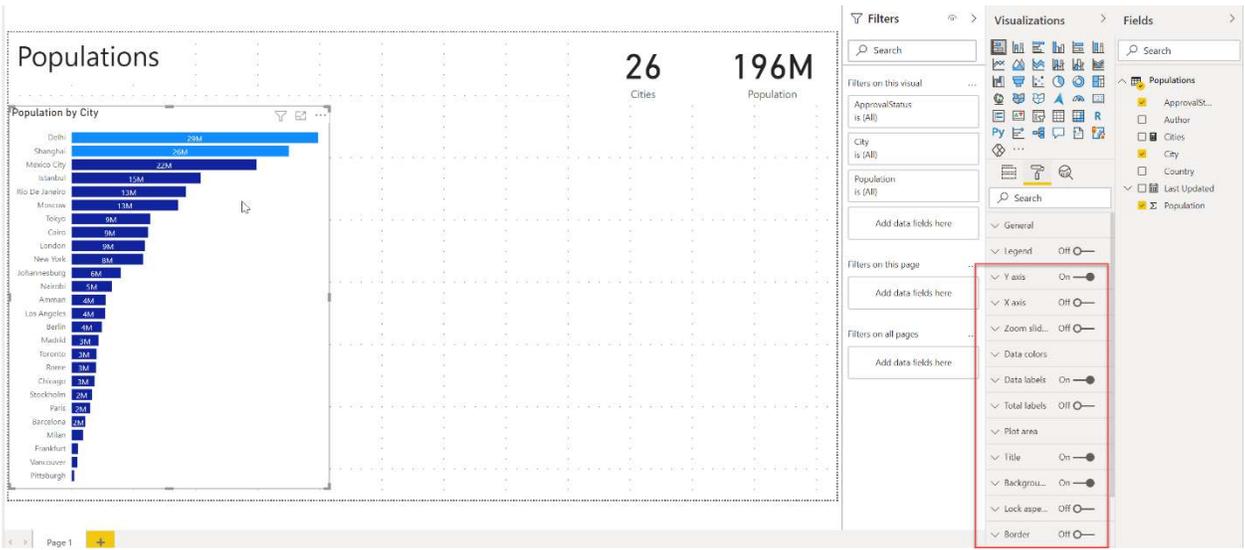
5. Add a bar chart to show Cities by population. The field named **Title** doesn't make sense from a naming perspective, so change it to **City** by double clicking on it in the Fields pane. It will be automatically updated everywhere. Click on an unused area of the report canvas and select **City** and **Population** from the Fields pane. Change the visual to a bar chart by clicking the bar chart icon in the **Visualizations** pane. Select **ApprovalStatus** from the Fields pane to add it to the visual. If it is not automatically added in the Legend section, drag it there. Resize the visual accordingly.



6. Change some of the visual properties. We want to show the population as a number on the chart, and as such, don't need some of the axes. In addition, we want to turn off the legend display as it takes up space, and the simple colour code can be determined by hovering over the data bars.

With the visual selected, select the Paint Roller icon from the Visualizations pane. Make the following changes:

1. Turn off Legend
2. Open the Y axis and turn off Title
3. Open the X axis and turn off Title
4. Turn off X axis
5. Turn on Data labels
6. Change Title to "Population by City"

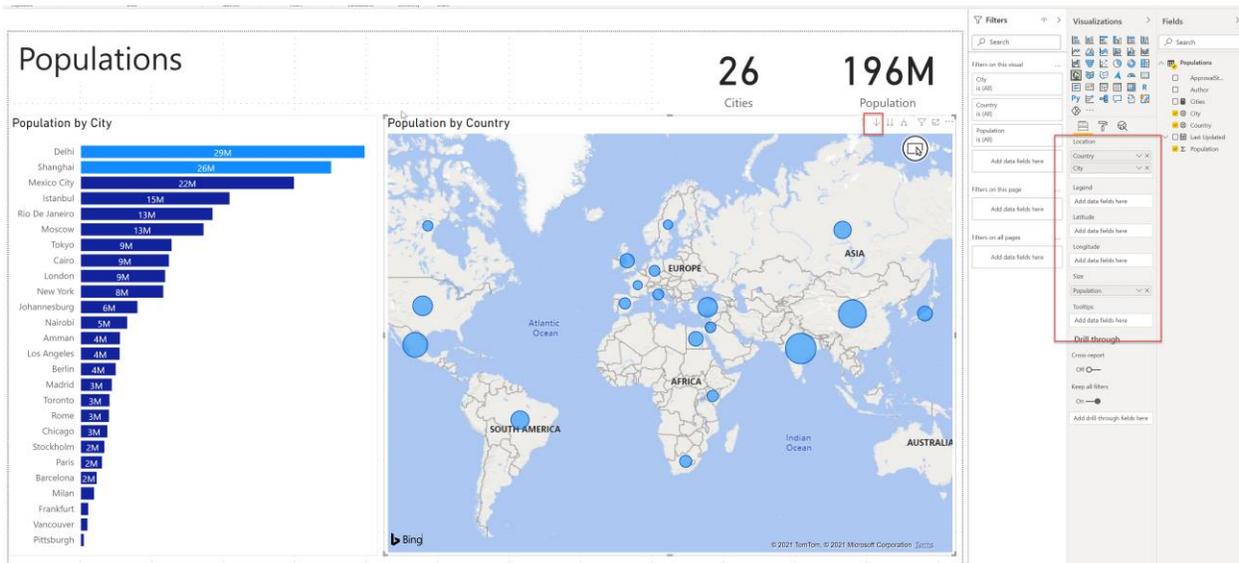


7. Add a map to the report. Select **Country** from the Fields pane (but don't "check" it). In the ribbon, under Column tools, select "Country" as its data category. Note the icon change in the Fields pane. Repeat this for the **City** field, selecting "City" as its data category.

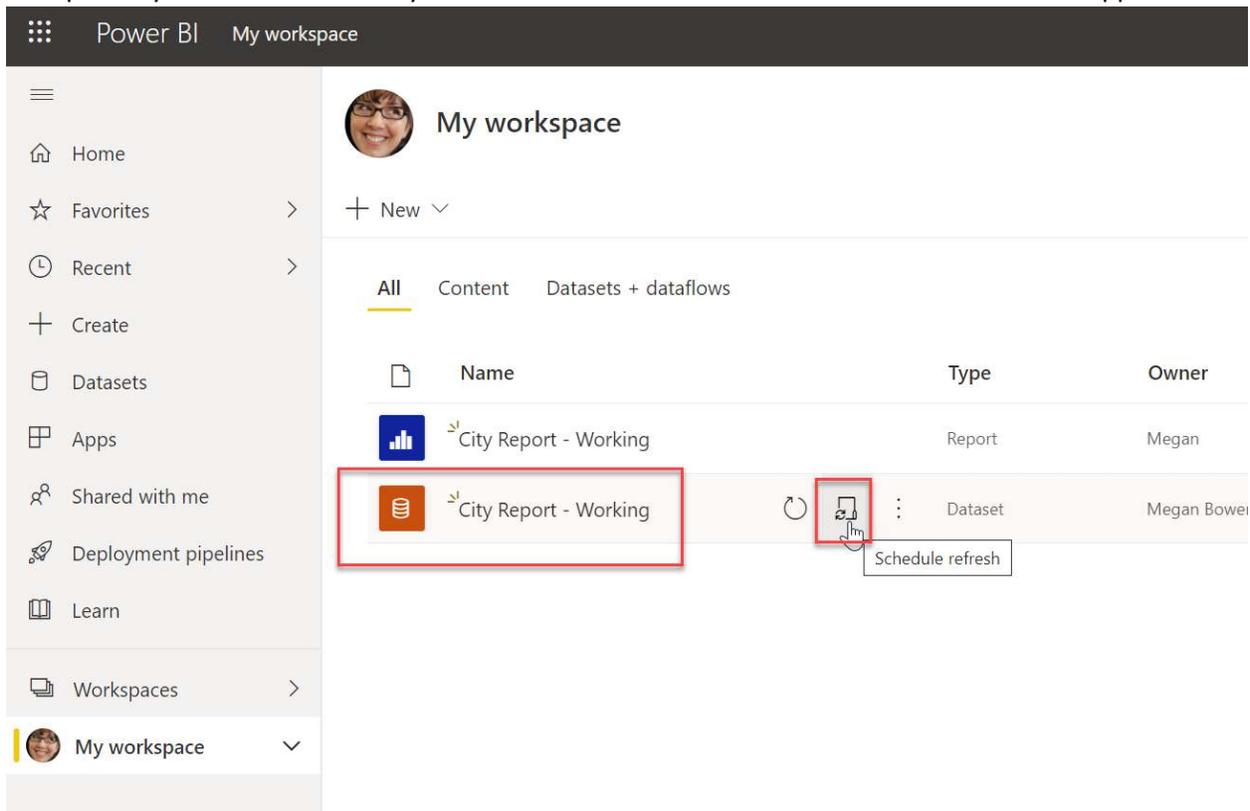
The screenshot shows the Power BI interface with the following elements highlighted:

- Ribbon:** The 'Column tools' tab is selected. The 'Data category' dropdown is set to 'City'.
- Fields Pane:** The 'City' and 'Country' fields are selected, indicated by a red box.
- Visualizations Pane:** The 'Location' section is highlighted with a red box.

8. Select an unused area of the report canvas and select **Country** and **Population** from the Fields pane. Next, select **City**, and then drag it from the Legend field to below **Country** in the location section. Resize the map accordingly.



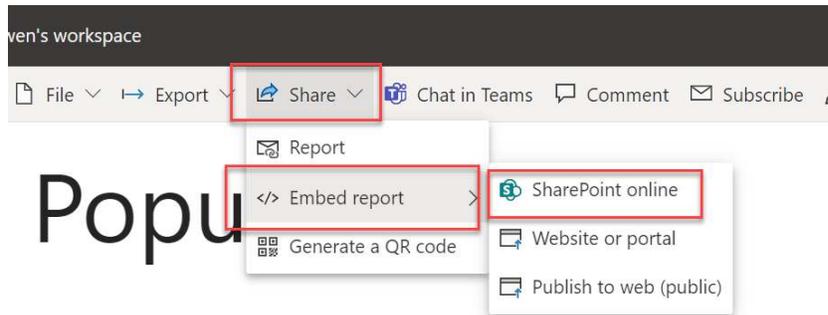
- Experiment with the map by turning on the drilldown icon (above) and then clicking on the circle for Canada or the US. Also experiment by selecting data bar for the various cities.
- Save the report and select the Publish button from the Home tab on the ribbon. When prompted, select “My workspace” as the destination.
- Open Power BI in a browser (<https://app.powerbi.com>) and navigate to your “My workspace”. Open the report by selecting it and ensure it is working properly.
- Set up a daily refresh. Hover over your dataset and select the **schedule refresh** button that appears.



- Open the **credentials** section and enter your credentials using OAuth2. Open the **Scheduled Refresh** section and turn on the option to **Keep your data up to date**. Your report will now be refreshed daily.

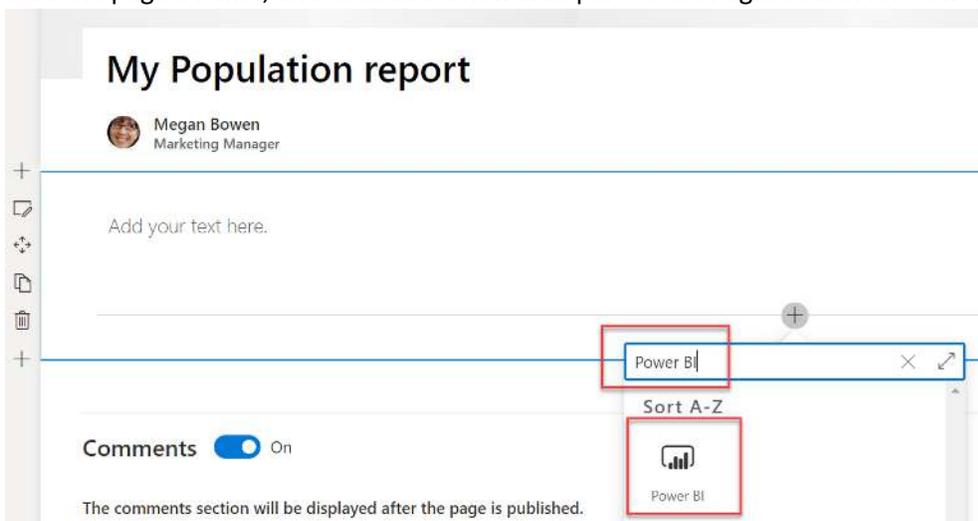
Embedding in SharePoint

1. Open the report in the Power BI service. From the button bar at the top of the report, select the **Share** dropdown, hover over **Embed report**, and select **SharePoint Online**.



Population by City

2. Copy the URL that pops up to your clipboard.
3. Open a SharePoint site, and then add a new Page.
4. Give the page a name, and then add a new web part to the Page. Search for Power BI in the add a web part list.



5. Select the Power BI icon, and then click **Add Report**. In the Web Part properties pane, paste in the URL that you copied in step 2
6. Turn off **Show Navigation Pane**
7. Publish your page, and experiment with the report on the page.

Congratulations! You have now completed all your hands-on labs!