

# WORKSHOP: Extending Business Central with model-driven apps

RAUL BOGAJO

ROBERTO CORELLA

V2: MAY -2025

# WORKSHOP: EXTENDING BUSINESS CENTRAL WITH MODEL-DRIVEN APPS

## Objectives

With the completion of the workshop, you will know:

- The basics of solutions management in Power Apps.
- How to create Dataverse tables and how to establish relationships.
- How to import data from Business Central Online to Dataverse using Power Platform Dataflows.
- How to create a simple (but cool...) model-Driven App.

## Scenario

CRONUS works with the job module and has defined machine type resources to assign them to their projects.

They need to keep track of the maintenance of these resources. Now they only want to register the date of maintenance, the person who performs it and if they found any incident during the revision process, but in the future, they want to make a much more exhaustive control and they need the application to evolve according to the operational needs.

For this reason, they decided to develop a model-driven app because it would cover this basic need and could be adapted to more complex needs in the future.

## Step 1 - Creating a solution.

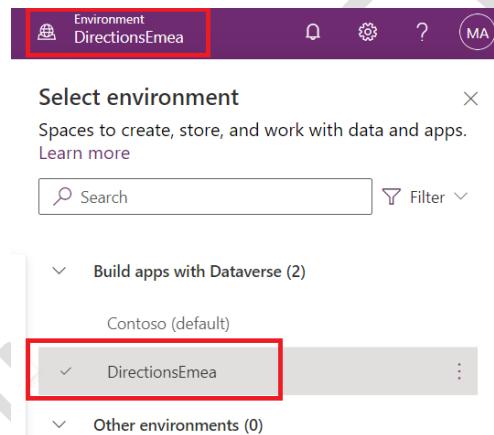
A Dataverse Solution is a way to package the different components that are part of a software unit developed with the Power Platform. Solutions are also used to move the packages between environments and to keep track about the authoring.

[Introduction to solutions - Power Apps | Microsoft Learn](#)

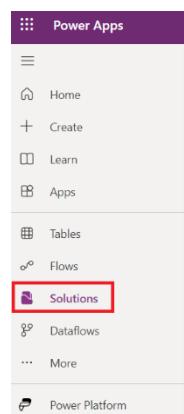
It is always a good practice, for the reasons mentioned above, to keep all the things together in a solution.

The first step of the workshops will be setting up our solution:

1. Navigate to [make.powerapps.com](#) and log in using your assigned credentials.
2. Select, on the top right corner, the [environment](#) DirectionsEMEA.



3. Select Solutions on the left pane.



4. Click on new solution:

The screenshot shows the 'Power Apps' interface. On the left, there's a sidebar with options like Home, Create, Learn, Apps, Tables, Flows, and Solutions. The 'Solutions' option is selected and highlighted with a purple bar. The main area is titled 'Solutions' and shows a table with two rows. The first row is 'Common Data Services Default Solution' with columns: Display name (Common Data Services Default Solution), Name (Crc53d3), Created (1 day ago), and Version (1.0.0.0). The second row is 'Default Solution' with columns: Display name (Default), Name (Default), Created (1 day ago), and Version (1.0). At the top of the main area, there are buttons for '+ New solution', 'Import solution', 'Open AppSource', 'Publish all customizations', 'See history', and a search bar.

5. Enter the display name and the name of the solution: Resource Management, preceded by the three initials of your name (this is just to identify your own solution inside the environment as other people are developing at the same time...):

The screenshot shows the 'New solution' dialog box. It has a purple header bar with the text 'New solution' and a close button 'X'. Below the header, there are two input fields. The first field is labeled 'Display name \*' and contains the value 'RBM Resource Management'. The second field is labeled 'Name \*' and contains the value 'RBMRessourceManagement'. There is also a third field labeled 'Publisher \*' which is currently empty.

6. When create new solution, we need to indicate the publisher. You can use the default publisher, but having a publisher that helps to identify the solutions that you or your company develops in the environment is the way to go. We will create a new one. Click on “New Publisher”:

## New solution

X

Display name \*

RBM Resource Management

Name \*

RBMResourceManagement

Publisher \*

Select a Publisher



+ New publisher

7. When create new solution, we need to indicate the publisher. You can use the default publisher, but having a publisher that helps to identify the solutions that you or your company develops in the environment is the way to go. We will create a new one. Click on “New Publisher” and enter the information:

- a. **Display Name:** your name
- b. **Name:** your name (no spaces allowed)
- c. **Description:** your name
- d. **Prefix:** your initials (the prefix will precede all the logical names of your objects and it will make easier to identify all the customizations developed by your company or team)

New publisher

Publishers indicate who developed associated solutions. [Learn more](#)

[Properties](#) [Contact](#)

Display name \*

RaulBogajo

Name \*

RaulBogajo

Description

Raul Bogajo as a publisher

Prefix \*

rbm

Choice value prefix \*

90925

Preview of new object name  
rbm\_Object

[Save](#) [Cancel](#)

In a real scenario this information should be something that identify a company or a team. Notice that you can enter additional information regarding contact.

8. When saved, we can select the publisher we just created as the publisher of our solution:

New solution ×

Display name \*  
RBM Resource Management

Name \*  
RBMRessourceManagement

Publisher \*  
 CDS Default Publisher (Crf3ae)  
Default Publisher for org245ddde6 (Def...)

More options ^

Installed on  
Sun Oct 29 2023

9. After selecting the publisher, notice that we can have versions of our solution, we can select a configuration page, and add a proper description. We leave the version as 1.0.0.0 and add some description and click on create.

New solution ×

Display name \*  
RBM Resource Management

Name \*  
RBMRessourceManagement

Publisher \*  
 + New publisher

Version \*  
1.0.0.0

More options ^

Installed on  
Sun Oct 29 2023

Configuration page

Description  
Machine Types Resources Management

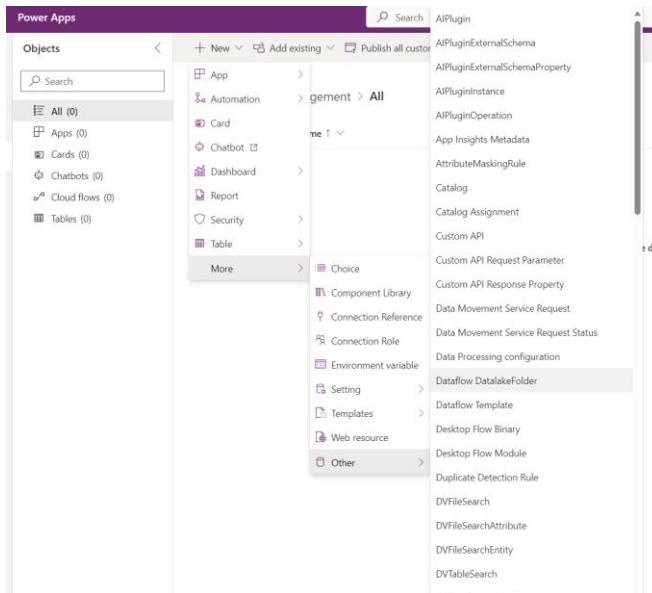
Package type (1)  
Unmanaged

Create Cancel

10. And the solution is ready! The next step is adding the components!

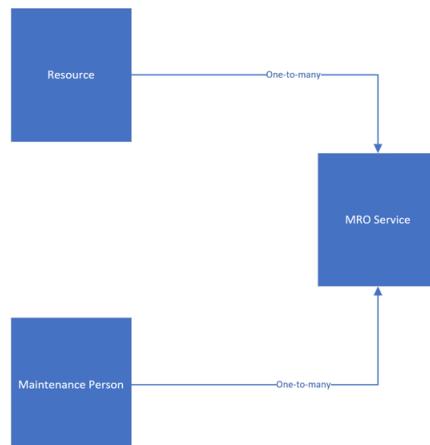
## Step 2 - Creating the data model: tables and relationships.

A solution can contains different components: tables, all kind of flows, different apps (canvas or model-driven), component libraries...



In the workshop we will create three custom tables in our solution to build our model-driven app:

- **Resource**: All the machine type resources created in Business Central. The records will be automatically inserted and updated with a Power Platform Dataflow.
- **Maintenance Person**: contains the data related to the maintenance staff.
- **MRO Service**: contains the data related to the maintenance actions planned and performed for the equipment.



Let's create the tables:

1. Select the solution and click on Edit.

The screenshot shows the 'Power Apps' interface with the 'Solutions' section selected. A solution named 'RBM Resource Management' is highlighted and has its 'Edit' button highlighted with a red box. Other solutions listed are 'Common Data Services Default Solution' and 'Default Solution'. The left sidebar includes options like Home, Create, Learn, Apps, Tables, Flows, Solutions, Dataflows, More, and Power Platform.

2. Select New -> Table -> Table.

The screenshot shows the 'Power Apps' interface with the 'Objects' section selected. The 'New' dropdown menu is open, and the 'Table' option is highlighted with a red box. Other options in the dropdown include 'App', 'Automation', 'Card', 'Chatbot', 'Dashboard', 'Report', 'Security', and 'Table from external data'. The left sidebar lists categories like All (0), Apps (0), Cards (0), Chatbots (0), Cloud flows (0), and Tables (0).

3. On the left pane, enter “Resource” as Display name and “Resources” as plural name. Also consider adding text in the description field that will help to identify what kind of information is stored on the table.

New table X

Use tables to hold and organize your data. Previously called entities [Learn more](#)

**Properties** Primary column

**Display name \***

**Plural name \***

**Description**

Enable attachments (including notes and files) <sup>†</sup>

[Advanced options](#) ▾

4. Click on Primary Column. The Primary Column is important as is the value that is used in all list views, when the table is related, to click through and navigate to the record's form. In this case we will leave the default value, as the resource name is good enough to be the displayed value for the primary column:

New table

Use tables to hold and organize your data. Previously called entities [Learn more](#)

**Properties** Primary column

**Display name \***

**Description**

[Advanced options](#) ▾

**Schema name \***

**Column requirement \***

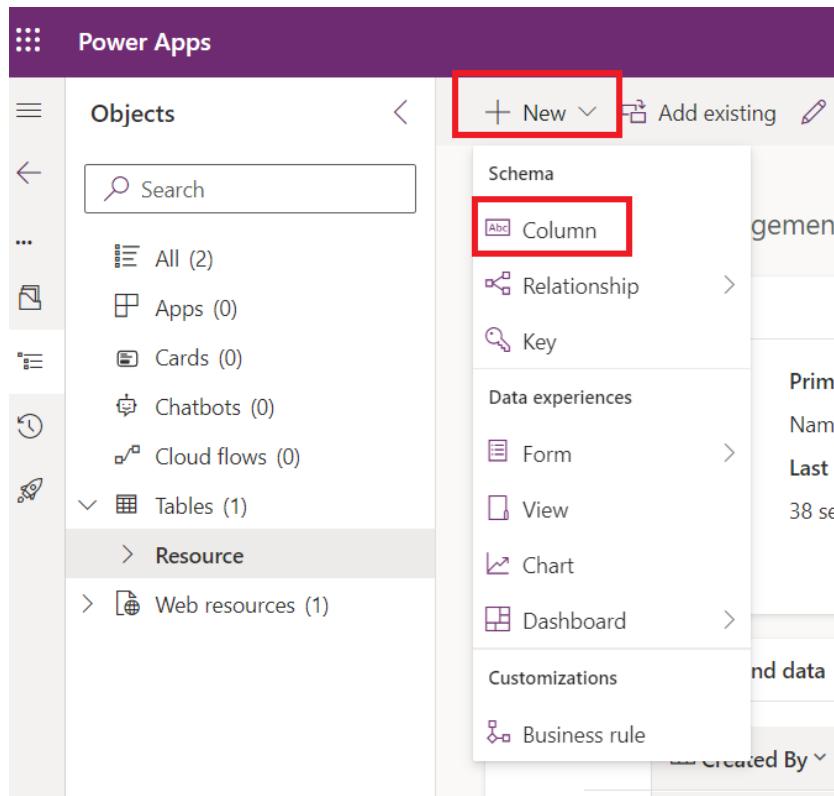
**Maximum character count \***

**Save** **Cancel**

Before saving, notice that the default value for the column requirement is “Business Required”. So, every record on the table must have this value informed.

The character length of the column can also be changed, but for this workshop, we will change the default value of 100 to 150 (100 is the length of the field “Name” of the Resource table in BC, but we will add the item No to have a perfect primary column!)

5. Click on save.
6. Once the table is saved, we need to add some fields. Select the table on the solution (it is the only table at this point...) and click on New -> Column.



7. A pane on the left will be open to define the new column:
  - a. **Display Name:** this is the name of the field as will be displayed. This column will identify the Resource No. as defined in Business Central.
  - b. **Description:** description related to the field. Is a good practice to enter valuable information here as it will be displayed, for example, in the flow designer of power automate when defining actions on the table.
  - c. **Data Type:** Dataverse tables have different column types. You can check them [here](#). For the item No, “Single Line of Text” will be ok!

- d. **Format:** Depending of the Data Type selected, [different formats](#) can be applied to the column that will be used to the UI to display the contents. Feel free to check the different options, but in this case, we will leave as plain text.
- e. **Behavior:** Some columns can be [calculated](#), and the behavior need to be changed. For this workshop, all the columns will have simple behavior, so we leave the default “Simple” option selected.
- f. **Required:** We can define the column as optional, business required, or business recommended. As we need to have a Resource No. in every record, we select the “Business Required” option.
- g. **Maximum Character Count:** Here we can define the length of the text field. For the Resource No., we can enter 20 (as is the length of the resource No defined in BC.)

The “No.” column, should look like this:

New column  
Previously called fields. [Learn more](#)

Display name \*

Description ⓘ

Data type \* ⓘ

Format \* ⓘ

Behavior ⓘ

Required ⓘ

Searchable ⓘ

Advanced options ^

Schema name \* ⓘ

Maximum character count \*

Input method editor (IME) mode \*

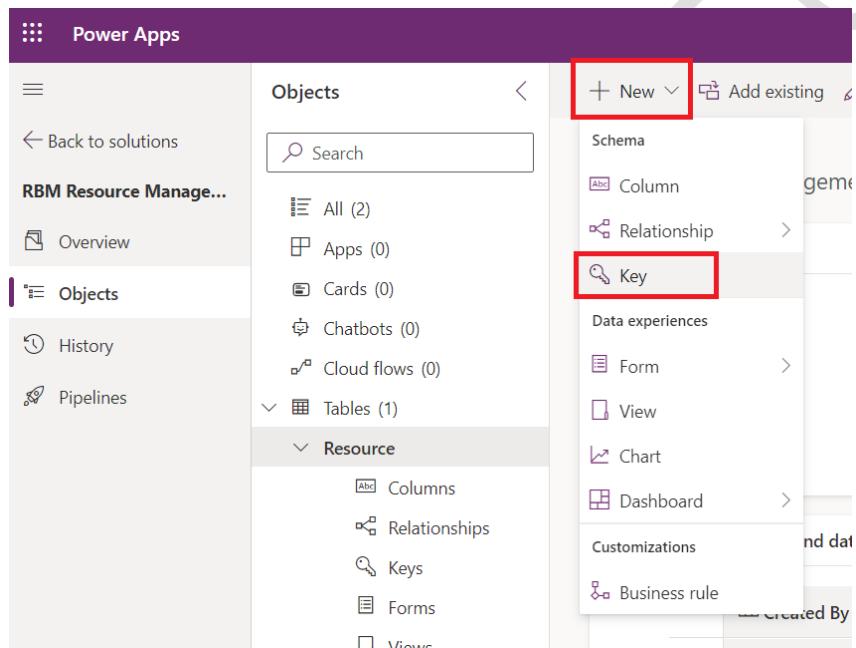
- 8. If everything is ok (looking as in the previous screenshot), click on save, and the column is ready!

9. Continue adding the rest of the fields as defined (none of them are business required):

Field Name	Data Type	Format	Length	Comment
Brand	Text	Text	100	Commercial brand of the resource
Model	Text	Text	100	Model of the resource
Date of acquisition	Date and Time	Date Only		Date when the resource was acquired.
Serial No.	Text	Text	100	Serial no. of the resource
Additional Information	Text	Text	500	Additional info related to the resource.

10. Finally, in the resource table we'll add an [alternate key](#). The key will be used later in the Power Platform Dataflow to manage the update of the existing records.

To create the key, click on New -> Key.



11. Name the key as “Resource No”, select the “No.” column and click on save.

The screenshot shows the 'Key' configuration screen in Business Central. It includes fields for 'Display name' (ResourceNo), 'Name' (rbm\_ResourceNo), and a 'Columns' section with various checkboxes. The 'No.' checkbox is checked. At the bottom are 'Save' and 'Cancel' buttons.

12. We need to create two tables more “Maintenance Person” and “MRO Service”. Create the tables as defined below:

The screenshot shows the 'Objects' page in Business Central. The 'Objects' menu item is highlighted with a red box. The 'Tables' item under 'Tables' is also highlighted with a red box. In the top navigation bar, the 'New' button is highlighted with a red box. The 'Table' item in the dropdown menu is also highlighted with a red box.

### Maintenance Person:

Display Name	Data Type	Format	Length	Comment
Name	Text	Text	100	<b>Primary Column</b>
Charge	Choice*			Technician, Supervisor, Manager
Email	Text	Email	100	
Phone No.	Text	Phone Number	100	

\* For the choice field (a list of values), you'll need to create a new choice field

New column  
Previously called fields. [Learn more](#)

Display name \* Charge

Description Current Charge

Data type \* Choice

Behavior Simple

Required Optional

Searchable

Selecting multiple choices is allowed

Sync with global choice?  Yes (recommended) Can be used in multiple tables, and will stay updated everywhere.  
 No Creates a local choice that can only be used in one table. People using it can add new choices.

Sync this choice with \* [New choice](#)

Default choice \* None

New choice

Display name \* Maintenance Person Charge

Label *	Value *	Sort
Technician	909,250,000	↑ ↓
Supervisor	909,250,001	↑ ↓
Manager	909,250,002	↑ ↓

+ New choice

Advanced options

### MRO Service:

Display Name	Data Type	Format	Length	Comment
Service Description	Text	Text	100	<b>Primary Column</b>
Service Type	Choice*			Maintenance, Repair, Overhaul
Date Planned	Date and time	Date Only		
Is completed	Yes/No			
Date Completed	Date and time	Date and time		

\* We need to create another [choice field](#) (a list of values). Same steps as the “Maintenance Person Charge” choice but creating different choice values.

Label *	Value *	Sort ▾
Maintenance	909,250,000	
Repair	909,250,001	
Overhaul	909,250,002	

+ New choice

Advanced options ▾

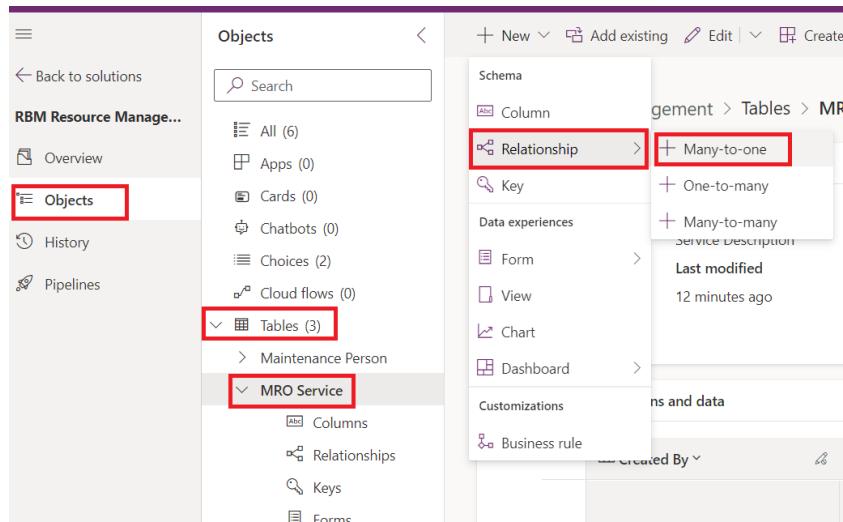
Notice that both new choices are created as objects, and added to the solution:

Display name	Name	Type	Managed	Last Modified
Maintenance Person Charge	item_maintenancepersoncharge	Choice	No	-
Maintenance Service Types	item_maintenanceservicetypes	Choice	No	-

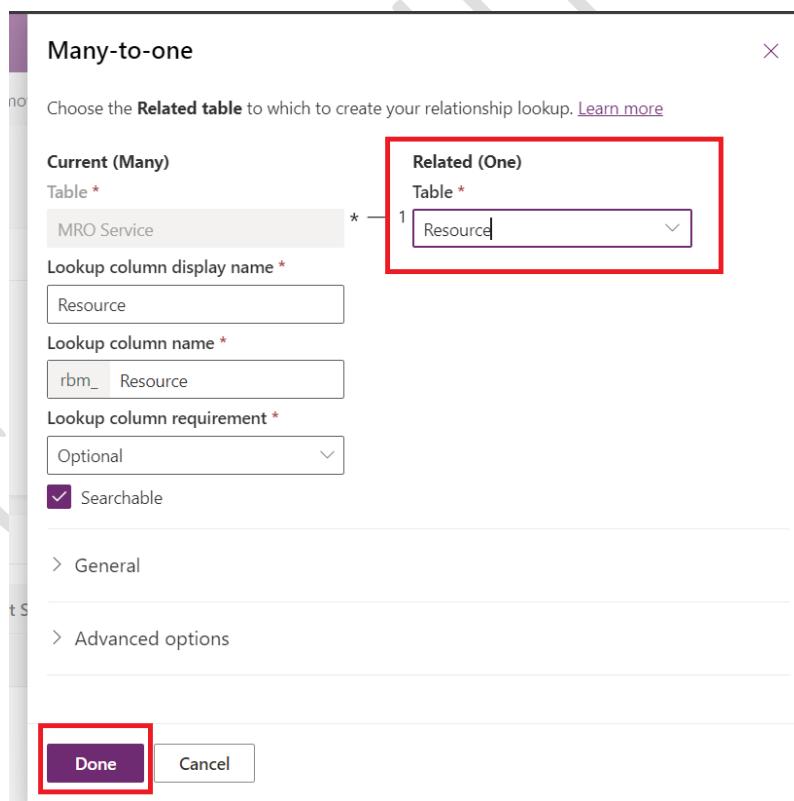
13. After all tables are created, we are going to establish the [relationships](#) between them.

As we need to establish a many-to-one relationship from the MRO Services to the Resource table, and another many-to-one relationship from MRO Services to the Maintenance Person, we can create both relationships from the MRO Services table.

Select the table and click on Relationship -> Many-to-one



14. Select “Resource” as the related table and click “Done”.



15. Repeat the previous step and select “Maintenance Person” at the related table.

16. After the relationships are created, look at the MRO Services columns. Two new columns, lookup type, are automatically added to the table:

Maintenance Person	:	rbm_MaintenancePerson	Lookup	No
Resource	:	rbm_Resource	Lookup	No

These columns created after defining the relationship, will allow you to select a record from the related table.

And, if you check the Maintenance or the Resource table, you will see the relationship defined as one-to-many:

The screenshot shows the Power Apps Relationships page for the 'Resource' table. The 'Relationships' section lists various relationships, with the last one, 'rbm\_MROService\_rbm\_Resource\_rbm\_Resource', highlighted in yellow. This relationship is defined as 'One-to-many' from the 'rbm\_MROService\_rbm\_Resource\_rbm\_Resource' table to the 'MRO Service' table.

Display name	Name	Related table	Relationship type	Managed	Customizable
Base Record ID	rbm_resource_DuplicateBaseRecord	Duplicate Record	One-to-many	No	Yes
Created By	lk_rbm_resource_createdby	User	Many-to-one	No	Yes
Created By (Delegate)	lk_rbm_resource_createdonbehalfby	User	Many-to-one	No	Yes
Duplicate Record ID	rbm_resource_DuplicateMatchingRecord	Duplicate Record	One-to-many	No	Yes
Entity instance	rbm_resource_PrincipalObjectAttributeAccesses	Field Sharing	One-to-many	No	Yes
Modified By	lk_rbm_resource_modifiedby	User	Many-to-one	No	Yes
Modified By (Delegate)	lk_rbm_resource_modifiedonbehalfby	User	Many-to-one	No	Yes
Name	rbm_resource_BulkDeleteFailures	Bulk Delete Failure	One-to-many	No	Yes
Object Id	rbm_resource_UserEntityInstanceDatas	User Entity Instance Data	One-to-many	No	Yes
Owner	owner_rbm_resource	Owner	Many-to-one	No	Yes
Owning Business Unit	business_unit_rbm_resource	Business Unit	Many-to-one	No	Yes
Owning Team	team_rbm_resource	Team	Many-to-one	No	Yes
Owning User	user_rbm_resource	User	Many-to-one	No	Yes
Record	rbm_resource_SyncErrors	Sync Error	One-to-many	No	Yes
Regarding	rbm_resource_AsyncOperations	System Job	One-to-many	No	Yes
Regarding	rbm_resource_ProcessSession	Process Session	One-to-many	No	Yes
Regarding Object Id	rbm_resource_MailboxTrackingFolders	Mailbox Auto Tracking Fol...	One-to-many	No	Yes
Resource	rbm_MROService_rbm_Resource_rbm_Resource	MRO Service	One-to-many	No	Yes

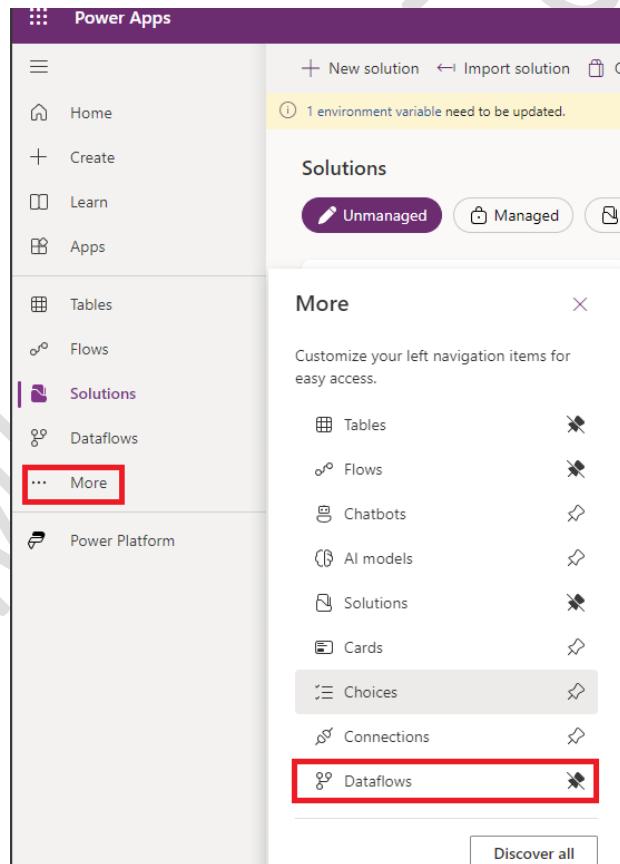
### Step 3 – Import the resources from Business Central Online to Dataverse

As the machine type resources are created in Business Central Online, we need to replicate the data to Dataverse so it can be used in our solution.

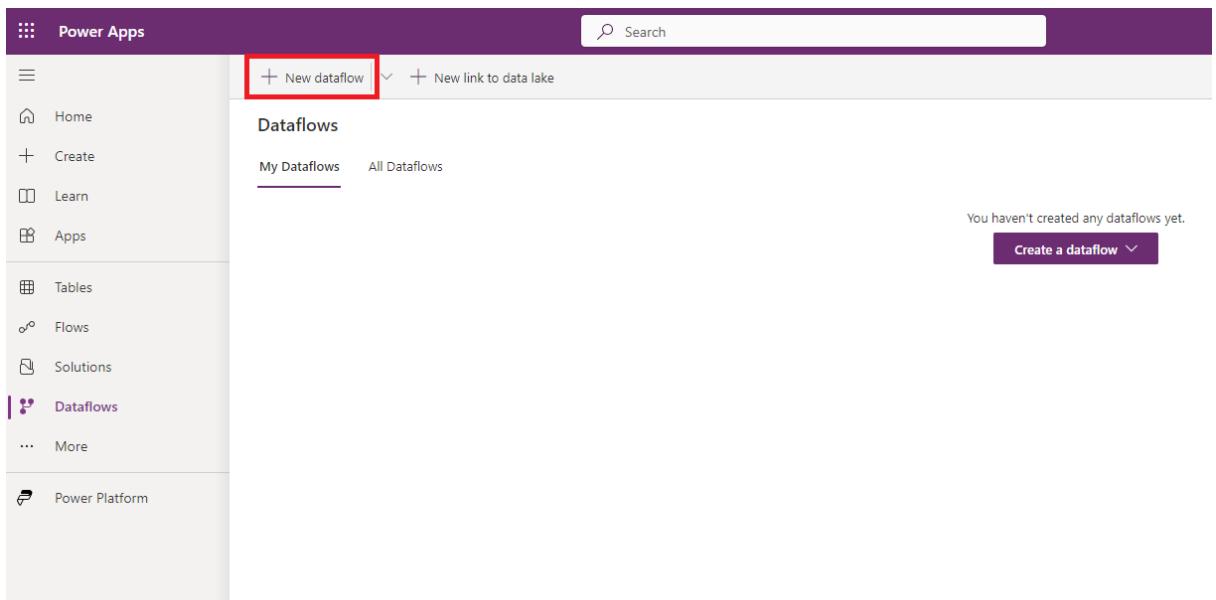
A Power Platform [Dataflow](#) is a good option to integrate data into Dataverse as we can connect to a lot of different sources, use Power Query to transform the data and storage the result of the transformation in a Dataverse table.

To create the dataflow that will populate the resource data from Business Central to the Resource table in Dataverse, we can follow the next steps:

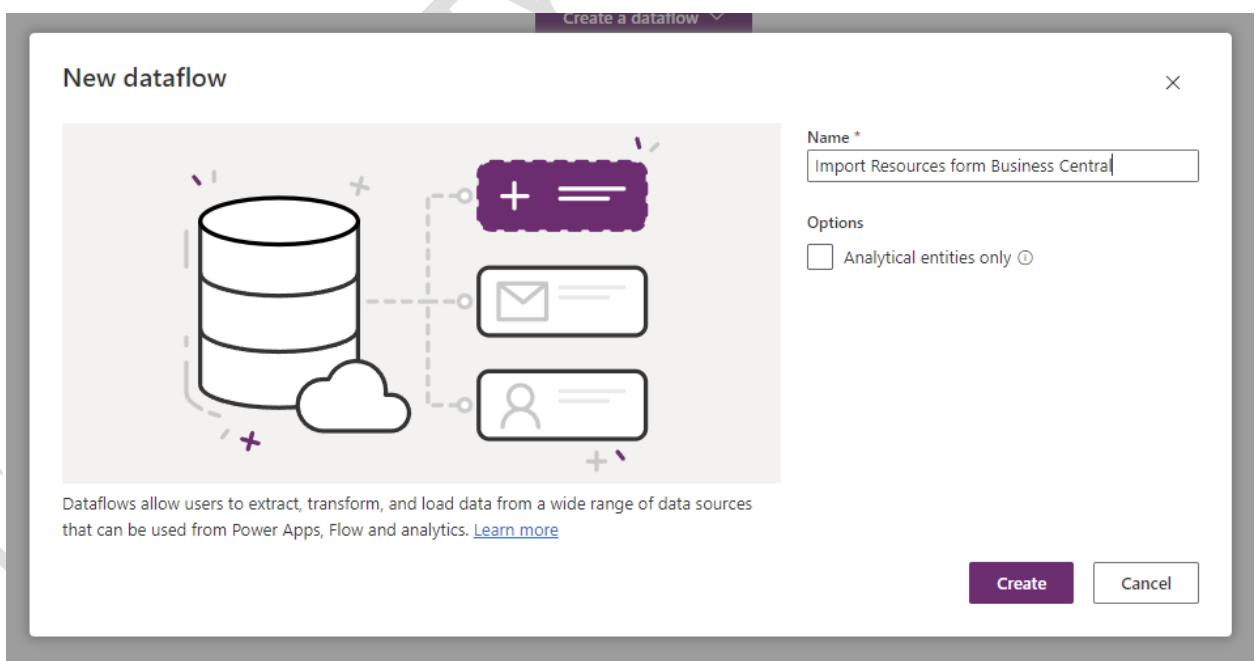
1. At this moment, we cannot create a dataflow inside the solution. We need to create it outside, and we will add it later. We need to navigate back to the home of make.powerapps.com and click on Dataflows on the right pane (notice that you can pin the element)
- 2.



3. Click on “New dataflow”



4. A pop-up displays and we need to enter the name of our dataflow. It should be something that describes what the dataflow will do (for the lab, precede the name of the dataflow with your initials). When the name is set, click on “Create.”



5. Now we can select the connection from different data sources. At this moment, Business Central Online does not have a native connector (as does for analytical dataflows defined in the Power BI service), so we will connect using OData.

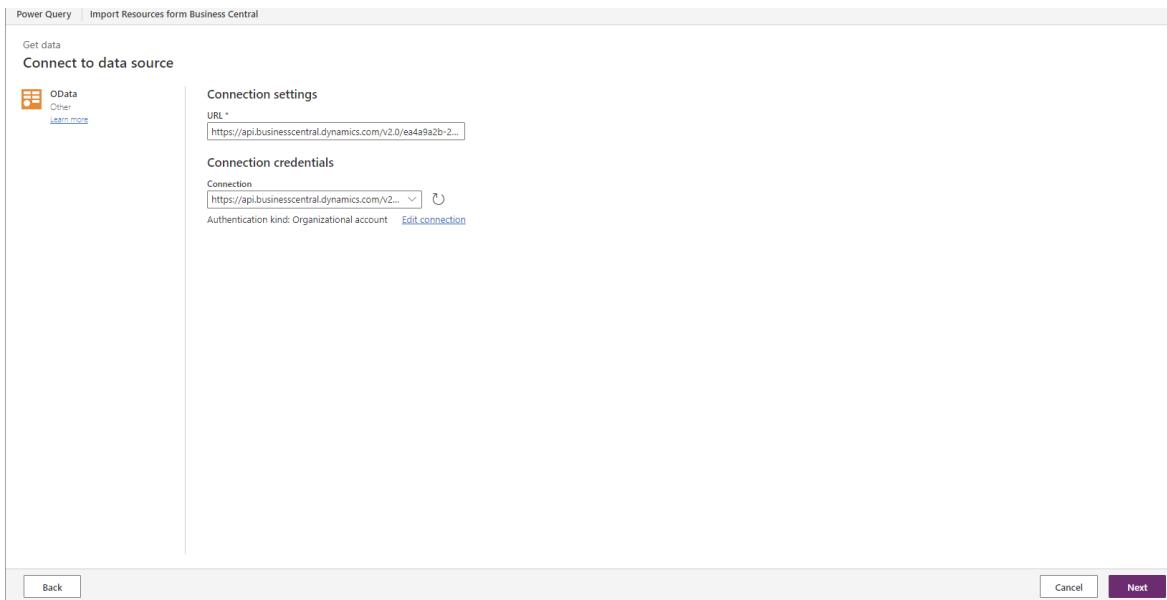
Select the OData connector.

6. We'll define the new connection using a custom endpoint to get the resources:  
[https://api.businesscentral.dynamics.com/v2.0/CommunityRocks/api/rbm/dataverse/v1.0/companies\(8f8b9802-b3d3-ee11-9050-6045bdc89e4e\)/resources](https://api.businesscentral.dynamics.com/v2.0/CommunityRocks/api/rbm/dataverse/v1.0/companies(8f8b9802-b3d3-ee11-9050-6045bdc89e4e)/resources)

And setting the Authentication as "Organizational Account":

7. If you signed in successfully the connection will be created, and we click on “Next” to open the Power Query Editor.

8.



9. The first thing to do in Power Query is filter the “Type” column to only get the machine type resources.

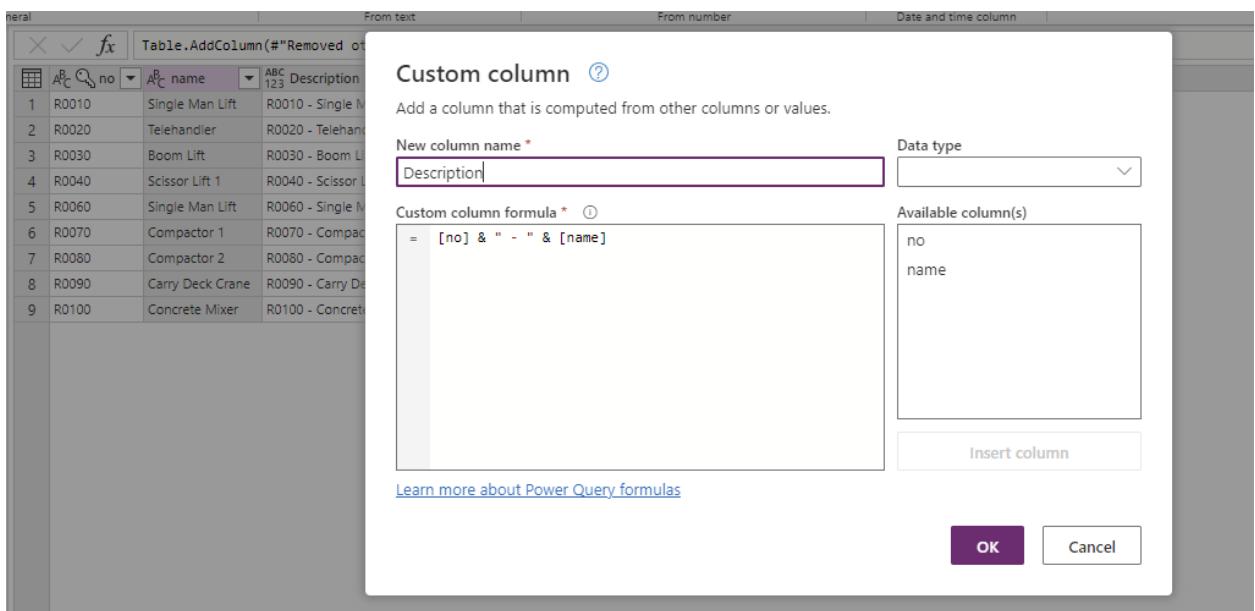
	A <sub>C</sub> no	A <sub>C</sub> type	A <sub>C</sub> name	A <sub>C</sub> address	A <sub>C</sub> city	1.2 directUnitCost	1.2 unitPrice
1	KATHERINE	Person				77	154
2	LINA	Person				92	185
3	MARTY	Person				69	139
4	R0010	Machine				0	0
5	R0020	Machine				0	0
6	R0030	Machine				0	0
7	R0040	Machine				0	0
8	R0050	Person				0	0
9	R0060	Machine				0	0
10	R0070	Machine				0	0
11	R0080	Machine				0	0
12	R0090	Machine				0	0
13	R0100	Machine				0	0
14	RESOURCE1	Person				0	0
15	RESOURCE2	Person				0	0
16	TERRY	Person				77	154

10. After filtering the records, we will remove the columns that we do not need. So, we select “no.” and “name” columns pressing ctrl and clicking on them, and right click on the mouse to choose “Remove Other Columns.”

The screenshot shows the Power Query interface with the 'Transform' tab selected. A context menu is open over a table of data, specifically targeting the 'no' and 'name' columns which have been selected by holding the Ctrl key. The menu options include 'Copy preview data', 'Remove columns', 'Remove other columns' (which is highlighted), 'Add column from examples...', 'Remove duplicates', 'Remove errors', 'Replace values...', 'Replace errors...', 'Merge columns', 'Change type', 'Transform columns', 'Group by...', 'Fill', 'Unpivot columns', 'Unpivot other columns', 'Unpivot only selected columns', and 'Move'. The main table view shows data from an OData feed, including columns like 'no', 'name', 'type', 'city', 'directUnitCost', and 'unitPrice'.

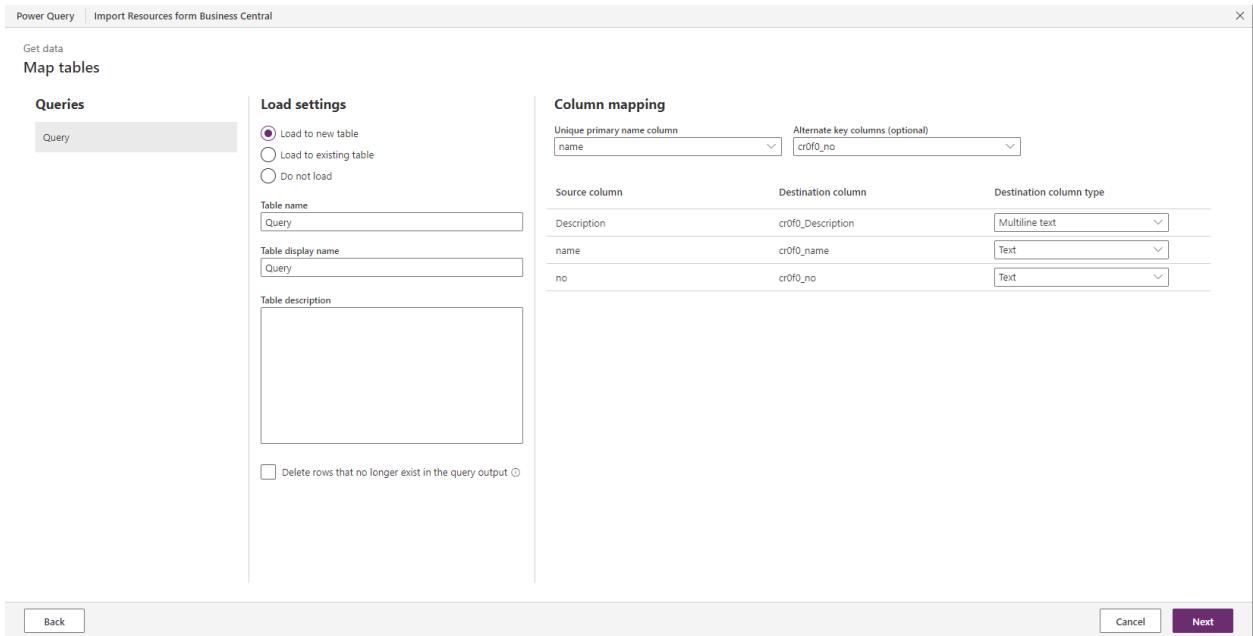
11. We will add a custom column that concatenates the No and the Name. We want to do that just because we will use this value in our primary column of the Resources table.

The screenshot shows the Power Query interface with the 'Add column' tab selected. A red box highlights the 'Custom column' button under the 'General' section of the ribbon. The main table view shows a query named 'Query' with the formula 'Table.SelectColumns(Source, {"no", "name"})'. The table data includes columns 'no' and 'name' with rows for KATHERINE, LINA, MARTY, R0010, and R0020.



12. This our last transformation, so we click on “Next”.

13. This our last transformation, so we click on “Next”, and we navigate to the “Map Table” screen.



14. We are going to Map the tables to the already created “Resource” table, so “Load setting” should look like that:

**Load settings**

- Load to new table
- Load to existing table
- Do not load

**Destination table**

rbm\_Resource

Table display name  
Resource

Table description  
Machine type resources imported from Business Central

Delete rows that no longer exist in the query output ⓘ

15. Once the table is selected, we can map the results of the query to the fields of the table:

### Column mapping

Source column	Destination column
(none)	rbm_AdditionalInformation
(none)	rbm_Brand
(none)	rbm_Dateofacquisition
(none)	rbm_Model
Description	rbm_Name
no	rbm_No
(none)	rbm_SerialNo

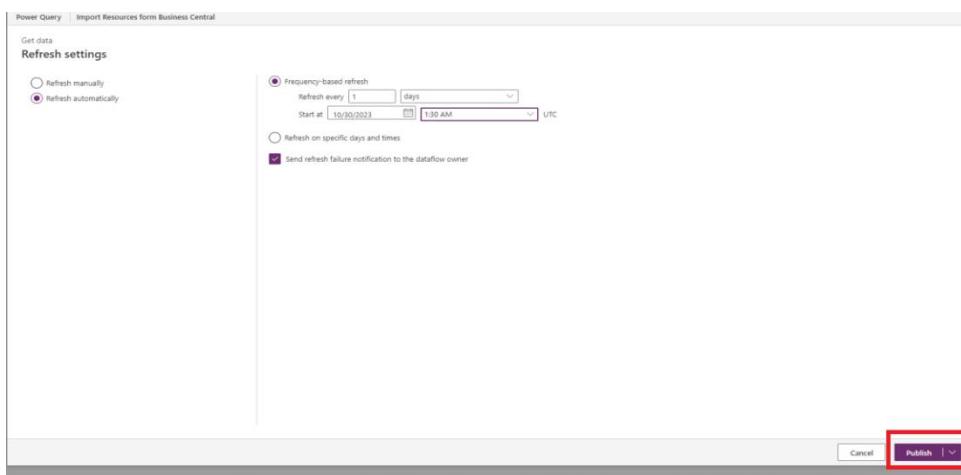
Notice that we selected the “Resource No.” key. This is the alternate key that we defined in earlier step where modeling the Resource table. As the resource no. is unique in BC, the definition of the key will allow to manage upserts (update and insert operations). If the key is not defined, the data will be duplicated after every execution of the dataflow.

When the mapping is ready, the dataflow is set, and we only need to define the refresh settings.

16. Select “Refresh Automatically” and establish the frequency as need:

The screenshot shows the 'Refresh settings' section of the Power Query interface. It includes two radio button options: 'Refresh manually' (unchecked) and 'Refresh automatically' (checked). Under 'Frequency-based refresh', it specifies 'Refresh every 1 days'. The 'Start at' field is set to '10/30/2023 1:30 AM UTC'. There is also an unchecked option 'Refresh on specific days and times' and a checked checkbox 'Send refresh failure notification to the dataflow owner'.

17. Everything is ready! So, we can publish the dataflow.



18. After publishing the dataflow, a data refresh will start automatically:

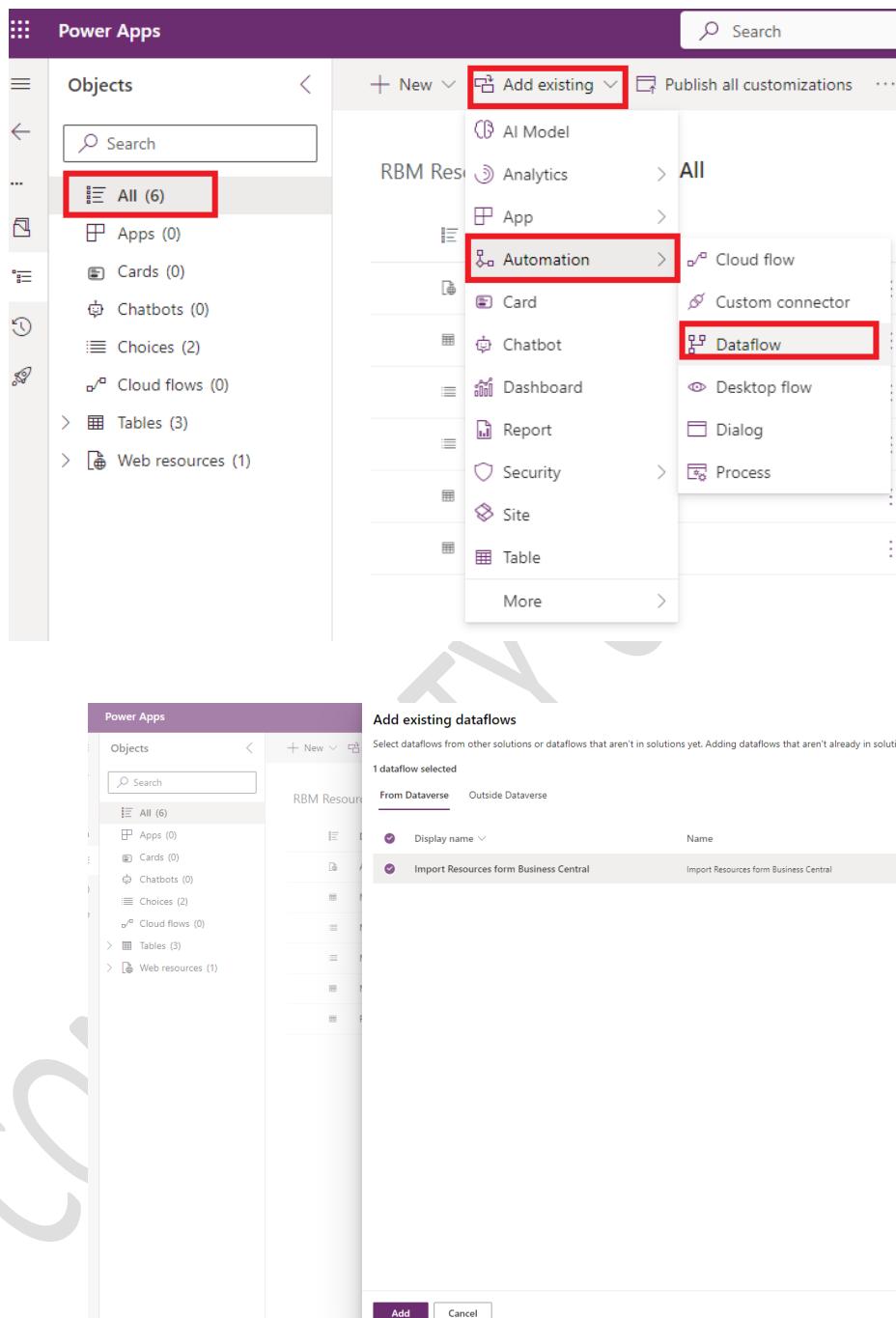
A screenshot of the "Dataflows" page. It shows a table with one row for "Import Resources from Business Central". The columns include Name, Type (Standard V2), Draft status (Published), Last published (10/30/2023, 7:33:52 PM), Last refresh (N/A), and Next refresh (In progress). The "Published" status is highlighted with a green checkmark.

19. When the process is ready, we can check that the data is created on the table:

A screenshot of the "Dataflows" page, identical to the previous one, showing the "Import Resources from Business Central" dataflow with its details: Standard V2 type, Published status, last published on 10/30/2023 at 7:33:52 PM, and the next refresh scheduled for 10/31/2023 at 2:30:00 AM.

A screenshot of the Power Apps portal. On the left, the "Objects" navigation pane is open, showing "Tables (3)" under "Resource". The main area displays the "Resource" table properties: Name (Resource), Primary column (Name), Description (Machine type resources imported from Business Central), Type (Standard), and Last modified (19 hours ago). Below this, the "Resource columns and data" grid shows a list of records with columns: Modified By, Name, and No\*. The data includes entries like MOD Administrator, R0040 - Scissor Lift 1, R0040; MOD Administrator, R0020 - Telehandler, R0020; etc., up to MOD Administrator, R0100 - Concrete Mixer, R0100.

20. The last step is to [add the recent created dataflow in our solution](#) (so we'll stick to the good practice in having all our components packaged in the solution):



## Step 4- Shaping the model-driven app.

As we have defined the data model and establish the relationship between the tables, the development of the model-driven app will be fast.

All tables created in Dataverse, standard or custom, have default views and forms associated with.

- Views: define how a list of records for a specific table will appear on the app
- Forms: define how the data for a record will be displayed and how the user will interact with it in the UI of the app.

The first thing to do is define (customizing the existing or creating new ones...) the views and forms in the tables before we design the app.

### Modifying views and forms in the tables

#### Views and forms in “Maintenance Person” table:

1. Open the solution.
2. Select the “Maintenance Person” table and click on Views.

The screenshot shows the Microsoft Power Apps portal interface. On the left, there's a navigation sidebar with 'Power Apps', 'Back to solutions', 'RBM Resource Management', 'Overview', 'History', 'Pipelines', and a 'Tables' section containing 'Maintenance Person', 'MRO Service', 'Resource', and 'Web resources'. The main area displays the 'Maintenance Person' table properties. Under 'Table properties', the 'Name' is 'Maintenance Person', 'Primary column' is 'Name', and 'Description' is 'List of the maintenance staff'. The 'Type' is 'Standard' and was last modified '21 hours ago'. To the right, there are tabs for 'Schema', 'Data experiences', and 'Customizations'. The 'Forms' tab is selected, showing a list with 'Views' highlighted and a red box drawn around it. Other options in the 'Forms' list include 'Relationships', 'Keys', 'Charts', and 'Dashboards'. At the bottom, there's a 'Maintenance Person columns and data' section with various filter and sort options, and a 'Update forms and views' button.

3. Select the “Active Maintenance People” view and click on “Edit.”

The screenshot shows the Microsoft Power Apps interface. On the left, there's a sidebar with 'RBM Resource Management' selected under 'Objects'. The main area is titled 'Objects' and shows a list of items: All (7), Apps (0), Cards (0), Chatbots (0), Choices (2), Cloud flows (0), Dataflows (1), Tables (3), and Web resources (1). To the right, a list of views for the 'Active Maintenance People' table is displayed, with the 'Edit' button highlighted by a red box.

#### 4. We'll add some columns to the view:

Select the Charge column and drag it next to the Name column.

The screenshot shows the Power Apps View editor. On the left, the 'Table columns' pane is open, showing various columns like 'Name', 'Created On', 'Charge', 'Created By', etc. A red arrow points from the 'Charge' column in this list to the header of the view, where the 'Name' and 'Created On' columns are listed. The view header also has a 'Charge' column listed.

Repeat the same action with the Email and “Phone No” fields.

The screenshot shows the 'Table columns' view for the 'Maintenance Person' entity. The left sidebar lists various system-related columns like 'Created By', 'Modified By', and 'Status'. The main area displays a grid icon and a message stating 'We didn't find anything to show here' with a note to try creating more records or removing filters.

5. Our view is ready, click on “Save and Publish”.

The screenshot shows the 'Save and publish' dialog for the 'Active Maintenance People' view. The 'Name' field is populated with 'Active Maintenance People'. The dialog also includes a 'Description' field and other standard save options.

6. Click on the back button to navigate to the solution page.

7. Confirm that you still have selected the “Maintenance Person” table and click on Forms.

The screenshot shows the Power Apps interface with the following details:

- Left sidebar:** Objects, Maintenance Person (selected).
- Top bar:** Search, New, Add existing, Edit, Create an app, Using this table, Import, Export, Advanced, Remove.
- Middle section:** RBM Resource Management > Tables > Maintenance Person. Table properties show Name: Maintenance Person, Primary column: Name, Description: List of the maintenance staff, Type: Standard, Last modified: 21 hours ago.
- Right sidebar:** Schema, Data experiences (Forms selected).

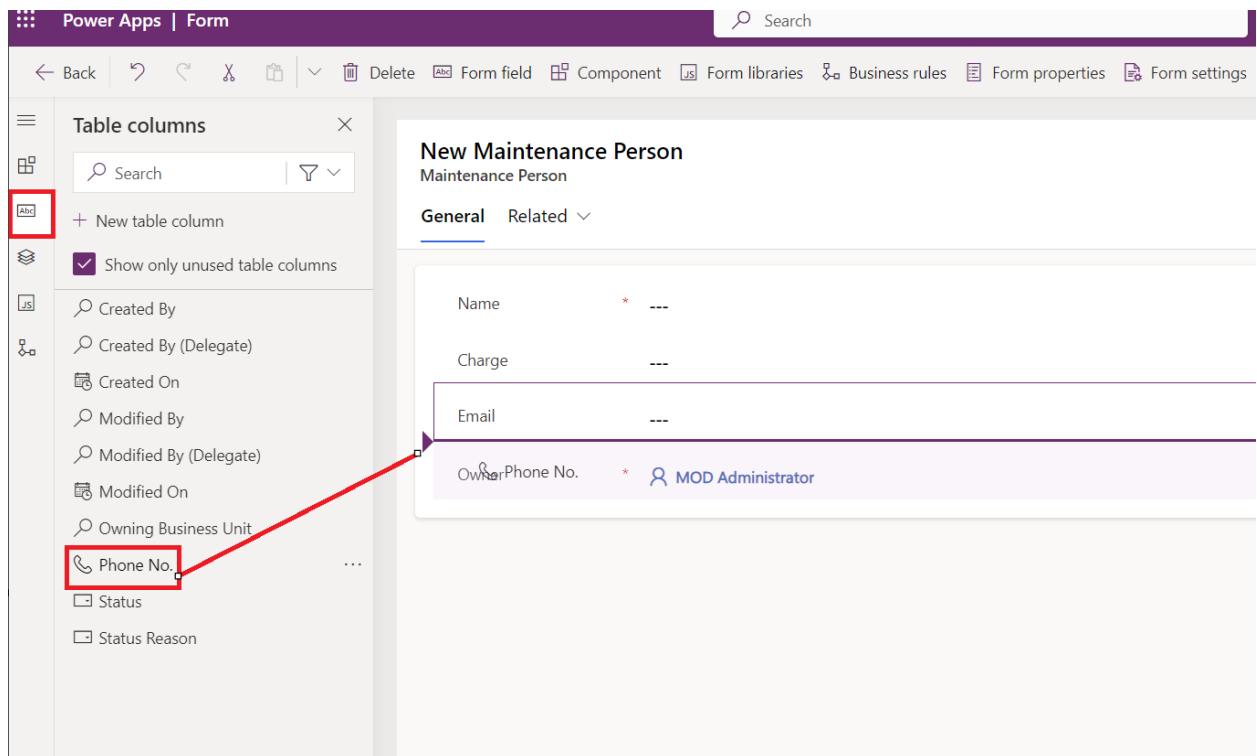
8. The Model-driven apps have different form types. We will edit the main form. Select it and click on edit.

The screenshot shows the Power Apps interface with the following details:

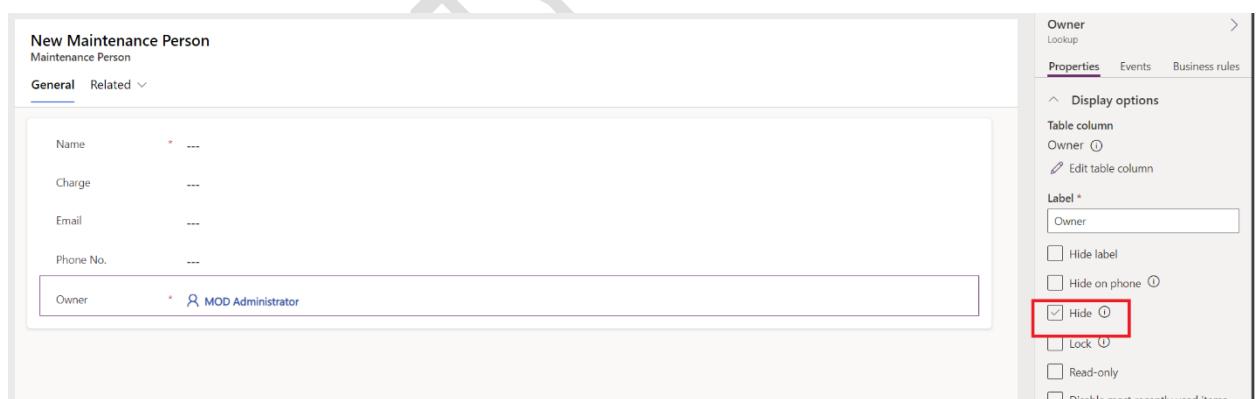
- Left sidebar:** Objects, Maintenance Person (selected), Forms (selected).
- Top bar:** Search, New form, Add existing form, Edit (highlighted with a red box), Turn off, Form settings, Advanced, Remove.
- Middle section:** RBM Resource Management > Tables > Maintenance Person > Forms. A list of forms is shown:

Name	Form type	Status	Managed
Information	Quick View	On	No
Information	Card	On	No
Information	Main	On	No

9. When the form is open in the edit mode, just drag the Charge, Email and Phone No. columns under “Name”:



10. As we do not need to see the Owner column on the form, just select it and hide it.



11. Our main form is ready. Is simple, (but effective... 😊).

The screenshot shows the Power Apps | Form interface. On the left, there's a sidebar titled 'Table columns' with a search bar and a list of columns: 'Created By', 'Created By (Delegate)', 'Created On', 'Modified By', 'Modified By (Delegate)', 'Modified On', 'Owning Business Unit', 'Status', and 'Status Reason'. The main area is titled 'New Maintenance Person' and 'Maintenance Person'. It has tabs for 'General' and 'Related'. Under 'General', there are four fields: 'Name' (with a red asterisk), 'Charge', 'Email', and 'Phone No.', each with a three-dot ellipsis button.

12. We recommend changing the default name to something more intuitive and click on “Save And Publish”.

The screenshot shows the 'Form properties' section. At the top right, there are buttons for 'Save a copy' and 'Save and publish' (which is highlighted with a red box). In the properties pane, under 'Display Name', it says 'Maintenance Person Main Form'. Under 'Description', it says 'Main form for maintenance person table.' There's also a 'Max Width (pixels)' input set to 1920 and a checked 'Show image' checkbox.

Extra: we are not going to use it on the workshop, but feel free to edit and experiment with the other forms defined for the Resource table (the quick view and the card...)

## Views and forms in “MRO Service” table:

1. Select the “MRO Service Table” on the solution and click on Views.

The screenshot shows the Power Apps portal interface. On the left, there's a navigation sidebar with 'Objects' selected. Under 'Tables', 'MRO Service' is listed. In the main content area, the 'Table properties' section shows details like Name (MRO Service), Primary column (Service Description), and Description (Maintenance, Repair or Overhaul services). To the right, there are tabs for 'Schema', 'Data experiences', and 'Customizations'. The 'Data experiences' tab is active, showing options for 'Forms', 'Views', 'Charts', and 'Dashboards'. The 'Views' option is highlighted with a red box. Below this, a grid shows 'MRO Service columns and data' with various filter and sort options.

2. Select and edit the “Active MRO Services” View and click on edit.

The screenshot shows the 'Views' list for the 'MRO Service' table. The left sidebar shows 'Objects' selected, and 'Views' is highlighted under 'MRO Service'. The main area lists six views: 'Active MRO Services' (selected and highlighted with a red box), 'Inactive MRO Services', 'MRO Service Advanced Find View', 'MRO Service Associated View', 'MRO Service Lookup View', and 'Quick Find Active MRO Services'. Each view has a status indicator ('On') and a 'Edit' button.

Name	View type	Status
Active MRO Services	Public View default	On
Inactive MRO Services	Public View	On
MRO Service Advanced Find View	Advanced Find View default	On
MRO Service Associated View	Associated View default	On
MRO Service Lookup View	Lookup View default	On
Quick Find Active MRO Services	Quick Find View default	On

3. Modify the view adding the columns: Date planned, Resource, Is Completed, Date Completed.

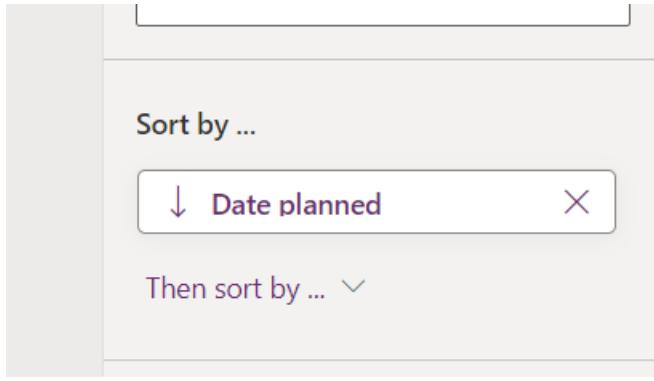
The result view should look like the screenshot below:

This screenshot shows the Power Apps View interface. The top navigation bar includes 'Power Apps | View', 'Search', and standard application controls. The main area displays a table with the following columns: Date planned, Resource, Service Description (sorted ascending), Is completed, Created On, and Date completed. Below the table, a message states 'We didn't find anything to show here' and 'This view does not have any data. Try creating more records or removing filters.' To the left, a sidebar titled 'Table columns' lists various fields under 'MRO Service' and 'Related'. Fields listed include Created By, Modified By, and Status.

4. We want to order the view by the “Date Planned” field. So just removed the current Sort on the left pane.

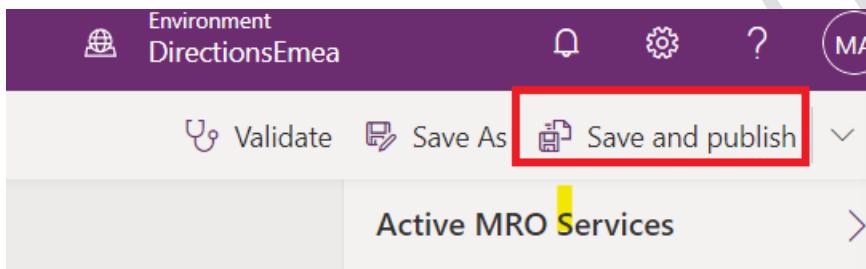
This screenshot shows the Power Apps View interface. The top navigation bar includes 'Search', 'Environment Directions Emea', and standard application controls. The main area displays a table with the following columns: Service Description, Is completed, Date completed, and Created On. Below the table, a message states 'We didn't find anything to show here' and 'This view does not have any data. Try creating more records or removing filters.' To the right, a sidebar titled 'Active MRO Services' shows settings for 'Name' (Active MRO Services) and 'Description'. A red box highlights the 'Sort by ...' section, which contains a dropdown menu with 'Service Description' selected. Other sections visible include 'Filter by ...' (Status is 'Active') and 'Edit filters'.

5. And just select “Date Planned” as the “Sort By” value:

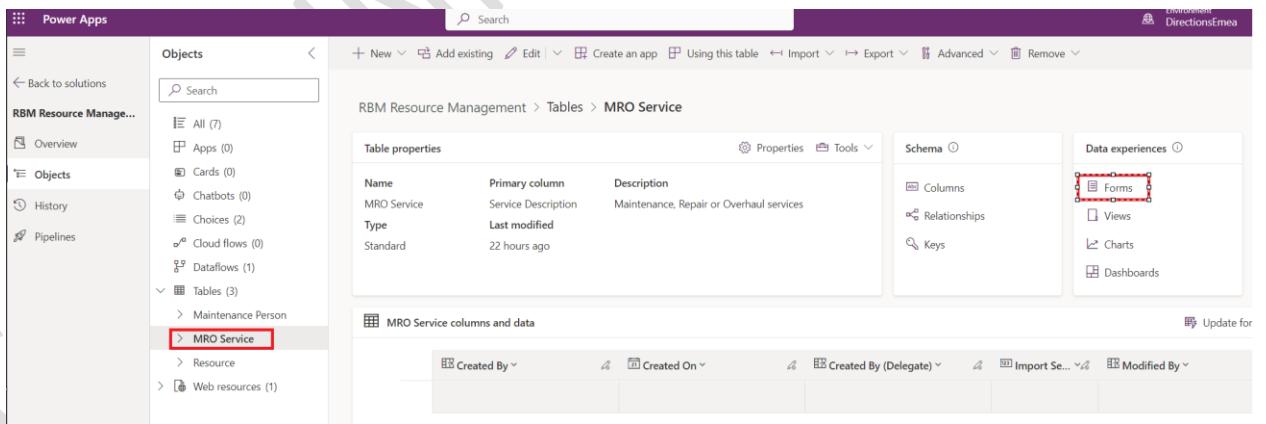


Notice that you can add more columns to define the order of the view.

- Everything ready for this view! Save and publish the changes!



- Go back to the solution. Select the “MRO service” table and click on “Forms.”



- We are going to customize the main form of this table. Select it and click on “Edit”.

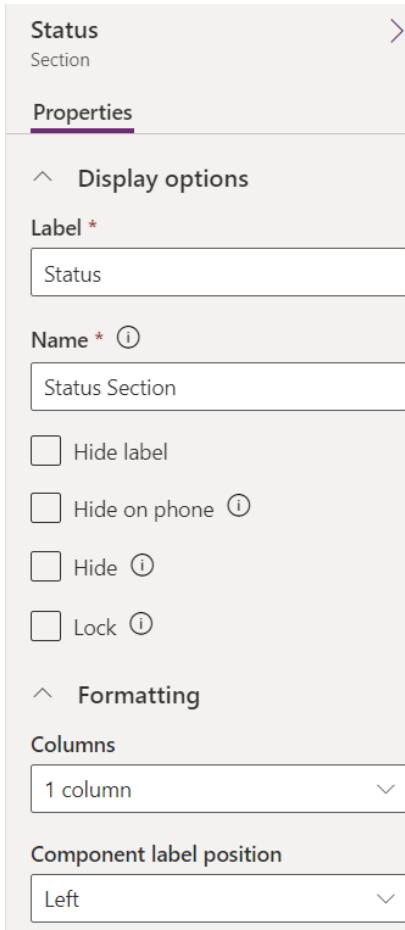
The screenshot shows the Power Apps interface with the 'Objects' section selected. In the center, there's a table listing forms under the 'Forms' category. The table has columns for Name, Form type, and Status. Three entries are shown:

Name	Form type	Status
Information	Quick View	On
Information	Main	On
Information	Card	On

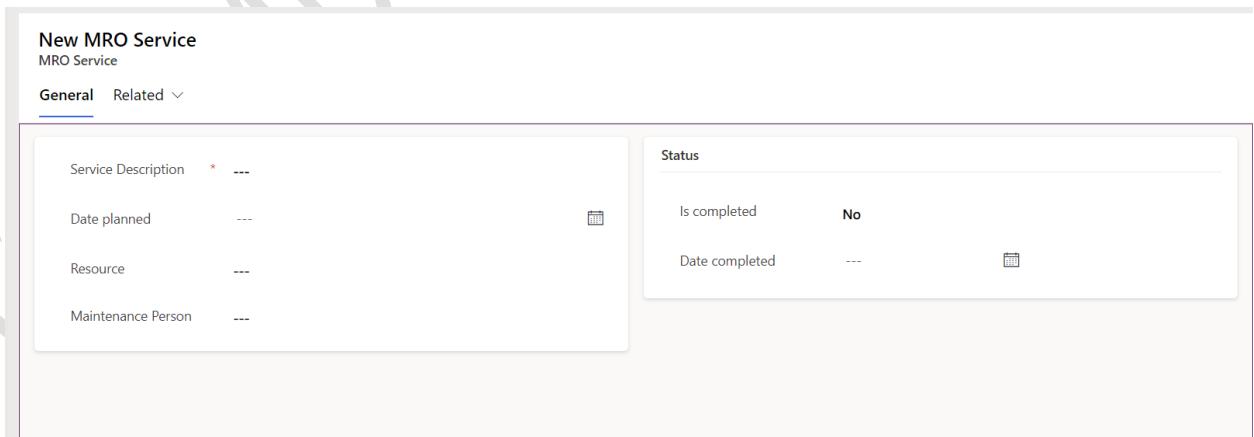
9. This time, we will use a 2 columns layout on the General tab. Click on the tab body and change the layout on the Properties pane:

The screenshot shows the Power Apps Form builder for a 'New MRO Service' record. The left sidebar lists table columns. The main area shows the 'General' tab with fields for 'Service Description' and 'Owner'. The properties pane on the right is open, showing the 'General' tab selected. A red box highlights the 'Layout' section under 'Formatting', which includes a dropdown for '1 column' and '2 columns'.

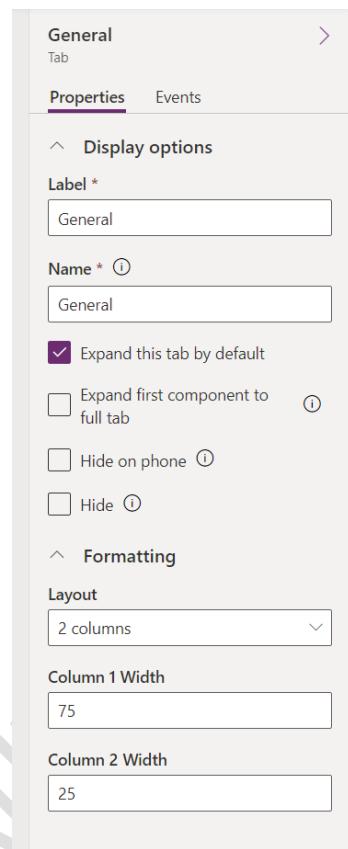
10. A new section will be added to the tab. Define the properties as follows:



11. With the already gotten knowledge try to add the fields to get a design like the screenshot below:



12. As we do not need the Status section to be this width, we can set up the width of the columns defined in the Tab. Select the tab properties and enter 75 in “Column 1 Width” and 25 in “Column 2 Width”.



13. Check the result! Looks better, is not?

The screenshot shows the 'New MRO Service' card in Business Central. The 'General' tab is selected. The card includes fields for 'Service Description' (with a required asterisk), 'Date planned', 'Resource', and 'Maintenance Person', each with a three-dot ellipsis button. To the right, there is a 'Status' section with a table:

Status	
Is completed	No
Date completed	---

14. We are ready with the MRO Service, change the form name in the form properties and save and publish!

### Views and forms in Resource table:

We are going to customize the views and the main for of the Resource table:

1. Select the Resource table on the solution and click on Views.
2. Select the “Active Resources” view and click on Edit.

The screenshot shows the Business Central interface with the following details:

- Left Sidebar:** Shows 'Objects' under 'RBM Resource Management...', with 'Tables' expanded and 'Resource' selected.
- Top Bar:** Includes 'New view', 'Add existing view', 'Edit' (highlighted with a red box), 'Turn off', 'Set as default view', and 'Advanced'.
- Table View:** Titled 'RBM Resource Management > Tables > Resource > Views'. It lists the following views:
 

Name ↑	View type
Active Resources	Public View default
Inactive Resources	Public View
Quick Find Active Resources	Quick Find View default
Resource Advanced Find View	Advanced Find View default
Resource Associated View	Associated View default
Resource Lookup View	Lookup View default

3. Modify the view to get the result showed below (you are an expert on views design!).

The screenshot shows the Power Apps View interface. On the left, there's a sidebar titled "Table columns" with a "Resource" tab selected. It lists various columns: No, Name, Brand, Model, Serial No., Date of acquisition, and a "View column" button. The main area displays a table with 10 rows, each containing a unique identifier (R0010 to R0100) and a corresponding name (e.g., "R0010 - Single Man Lift").

No	Name ↑	Brand	Model	Serial No.	Date of acquisition	+ View column
R0010	R0010 - Single Man Lift					
R0020	R0020 - Telehandler					
R0030	R0030 - Boom Lift					
R0040	R0040 - Scissor Lift 1					
R0060	R0060 - Single Man Lift					
R0070	R0070 - Compactor 1					
R0080	R0080 - Compactor 2					
R0090	R0090 - Carry Deck Crane					
R0100	R0100 - Concrete Mixer					

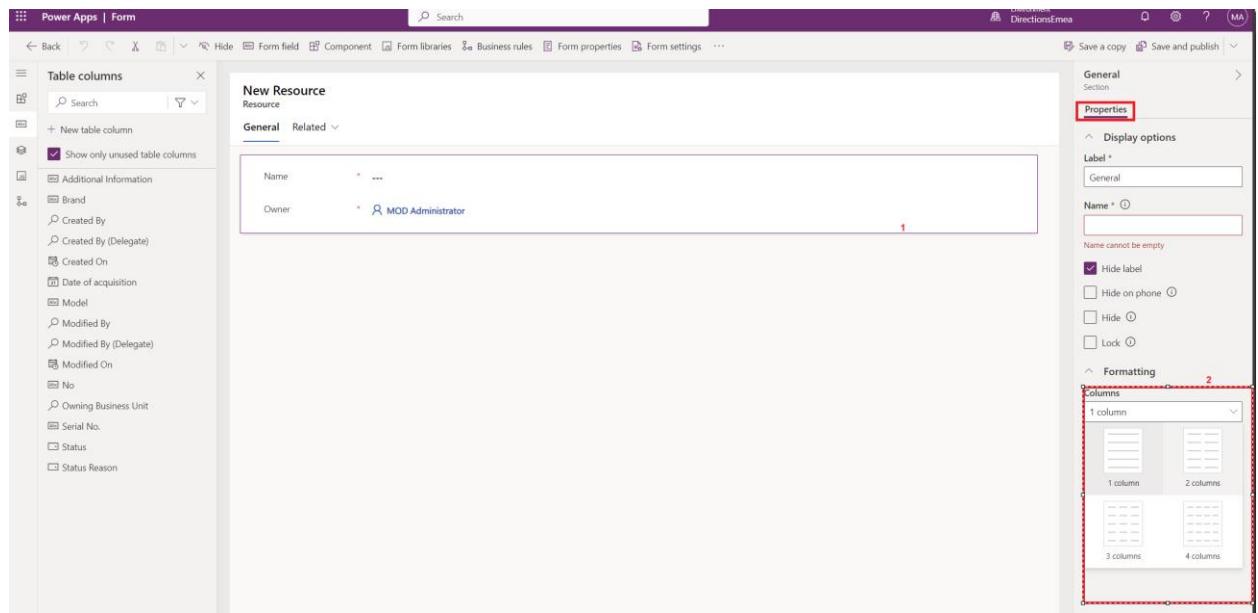
4. When you are ready, save and publish the modified view and go back to the solution.

5. Select the Resource table, click on Forms, select the main form, and click on Edit.

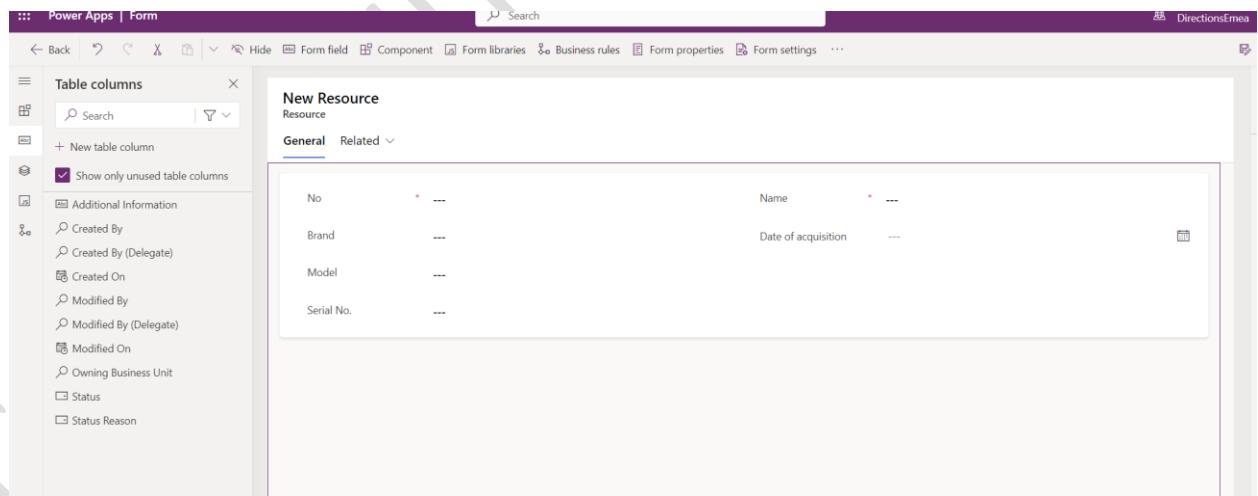
The screenshot shows the Power Apps Objects screen. The left sidebar has a "Objects" section with "Tables" expanded, showing "Resource" selected. Under "Resource", the "Forms" option is highlighted with a red box. The main area shows a list of forms for the "Resource" table, ordered by Name. The first form, "Information", is highlighted with a red box and has "Main" selected under "Form type".

Name ↑	Form type	Status
Information	Card	On
Information	Quick View	On
Information	Main	On

- We will pay a little more attention to this form. First, we will change the general section layout property to 2 columns. Click on the component, and change the value of the Columns property to 2 columns:

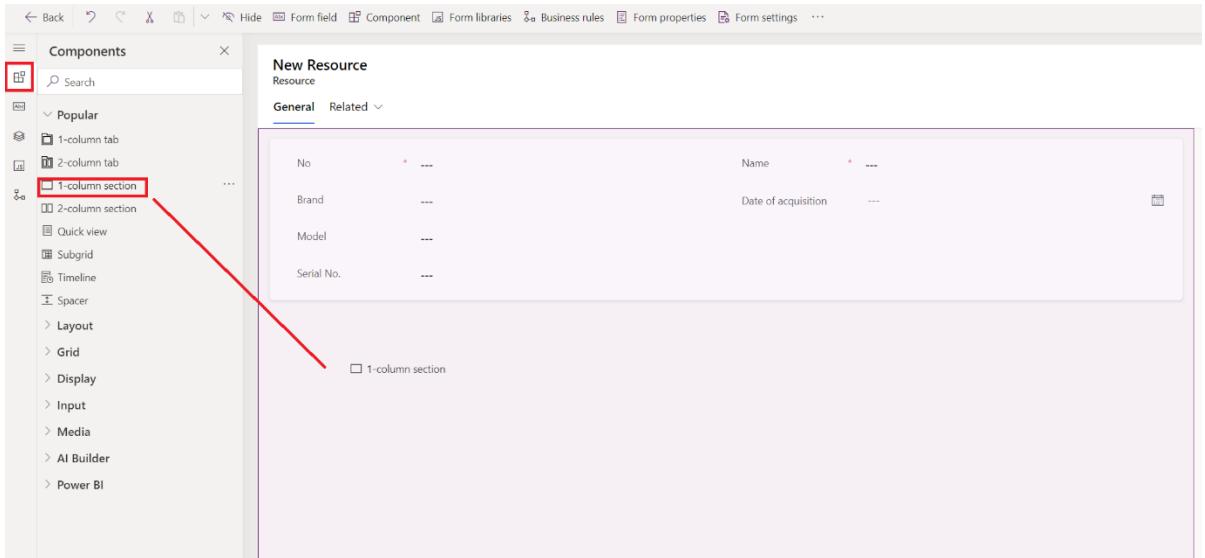


- Then add the columns to the General section. The result should be something like the next screenshot:

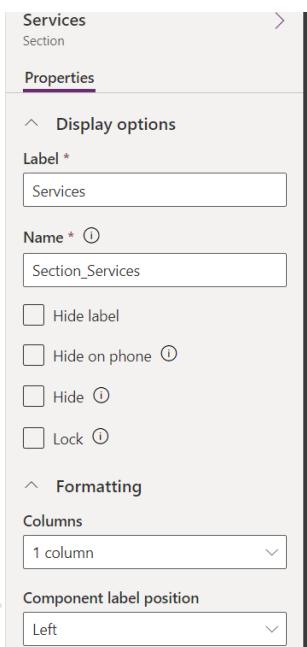


Do not forget to hide the Owner column!

- Then we will add a new section below the existing one. Click on Components, select 1-Column section and drag it below the existing one.

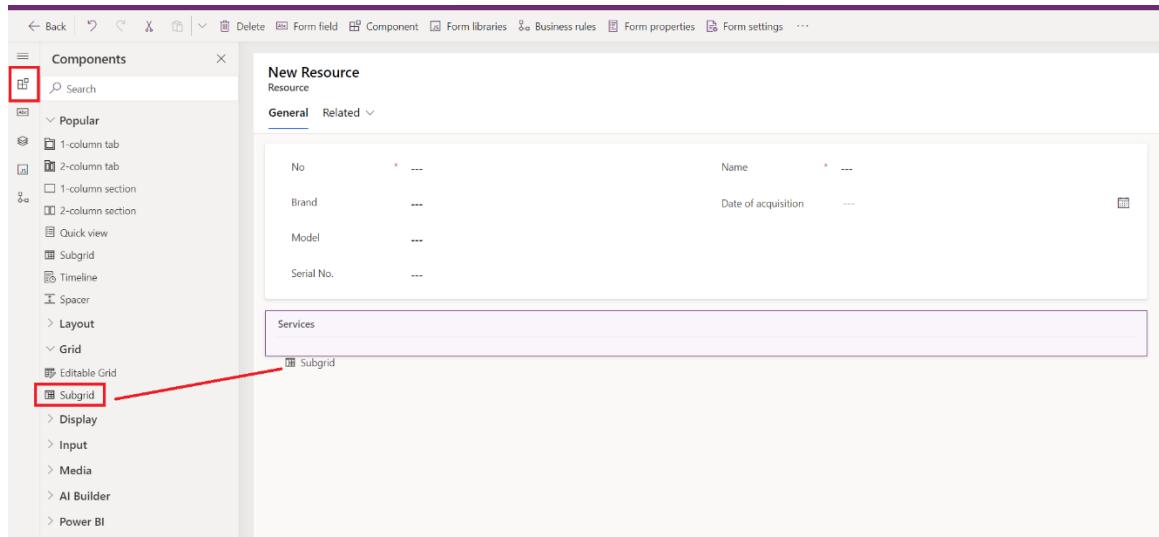


9. Change the value of the Label property to Services and enter Section services as the component name.

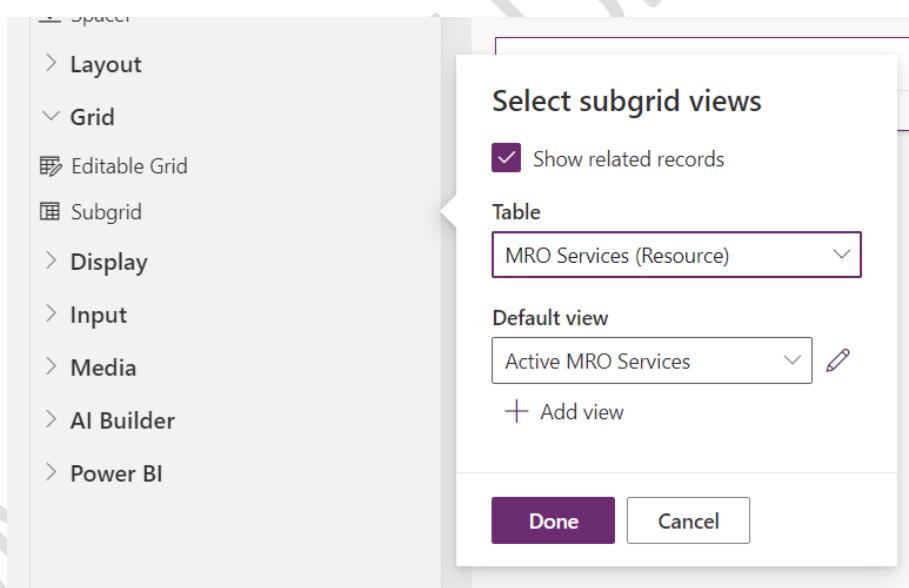


10. We need to show on the form all the services related to the resource displayed. To get this job done, we will add a sub grid to our services section.

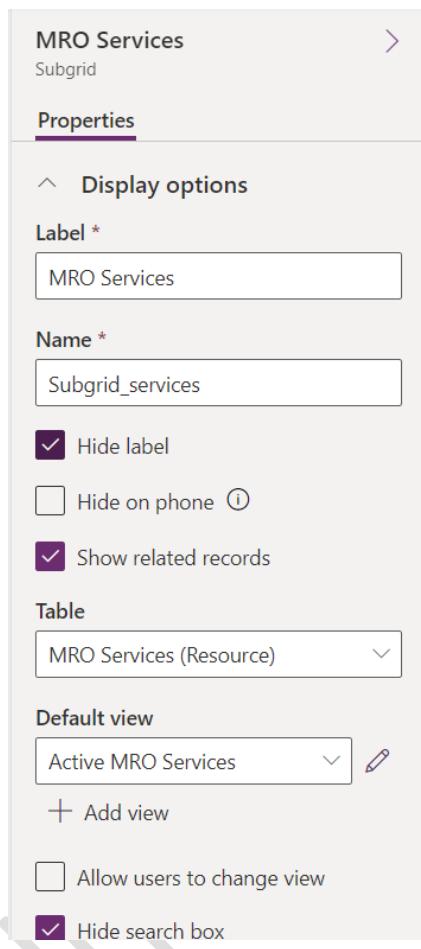
Select the subgrid component and drag it to the brand-new section.



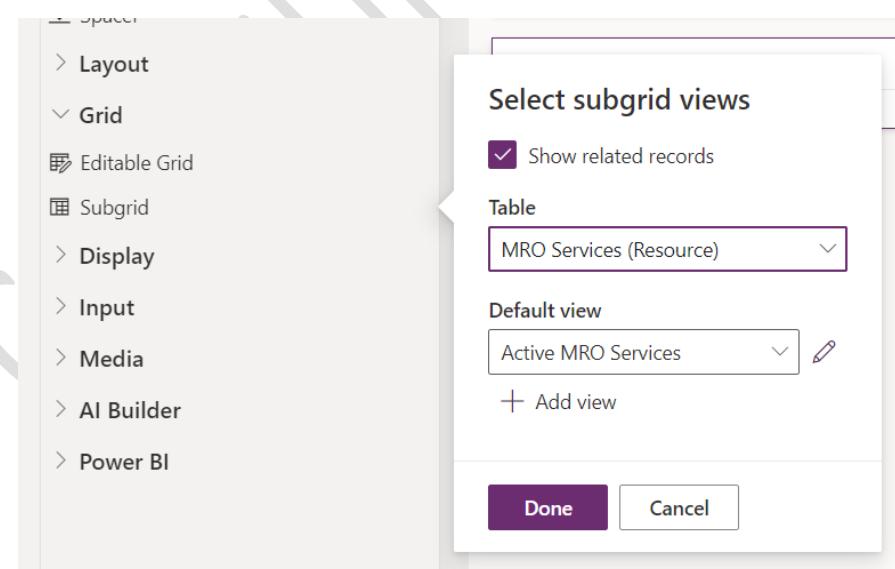
11. As we add the subgrid a pop up will appear to define the subgrid view. Click on “Show related records” to select only related tables to Resource. Select the “MRO Services” table, keep the default view as “Active MRO Services” and click on Done.



12. Change the label and name properties of the subgrid and activate “Hide Label.”



13. We are almost done! Change the form name in form properties...

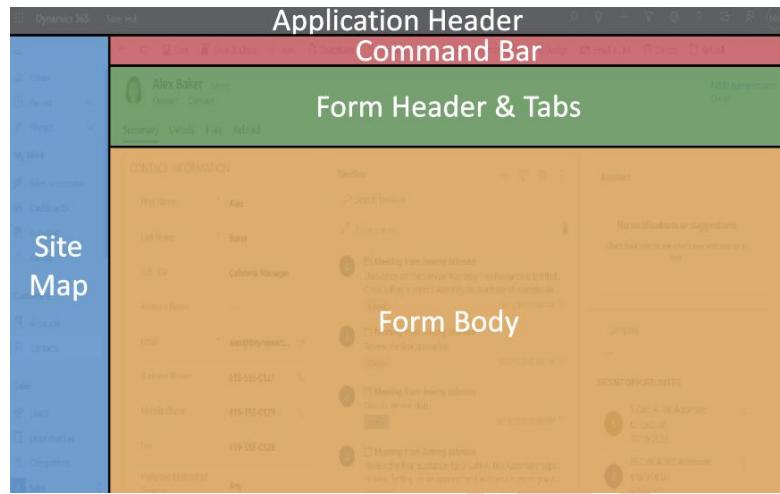


14. ...and Save and Publish!

## Shaping the model-driven app

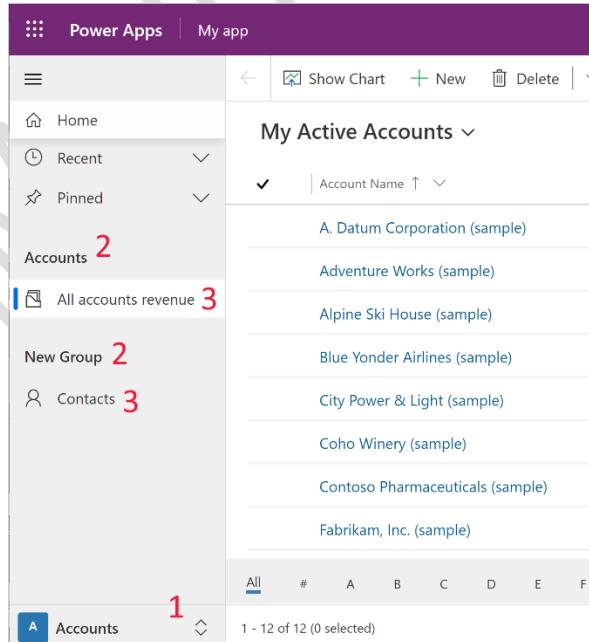
As we defined the views and forms for the table components of our app, the 90% of the work is done and we can create our model-driven app.

First, we will look at the structure of a model-driven App:



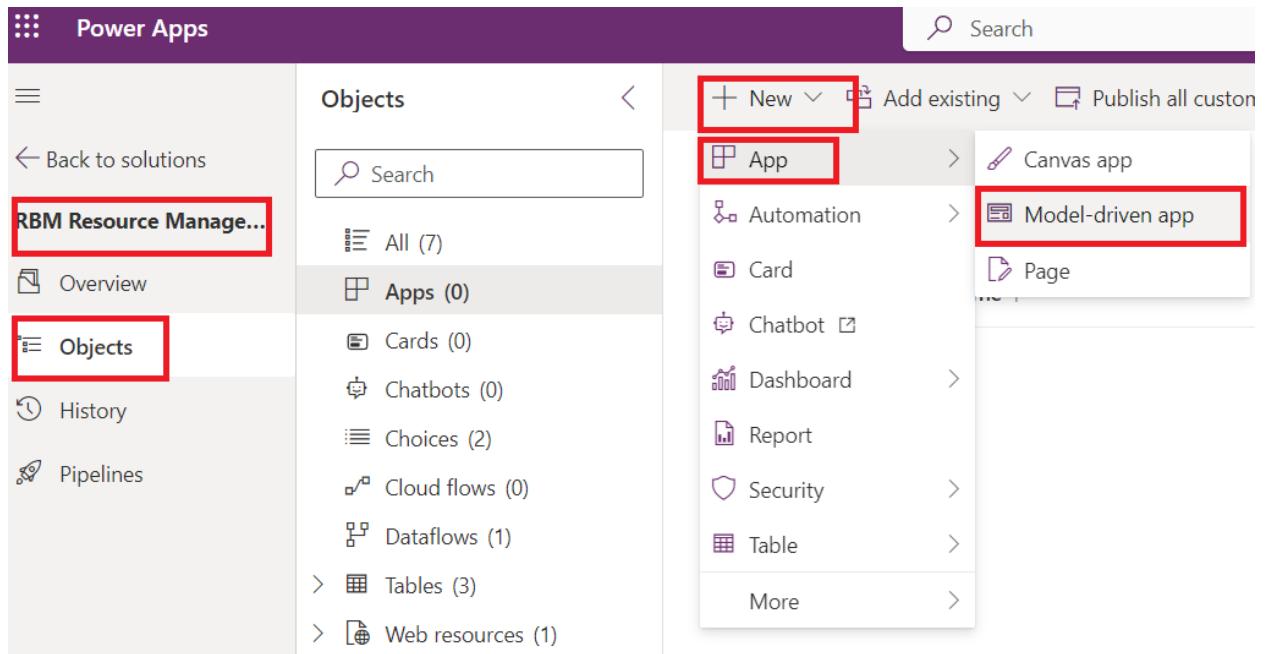
The app navigation is defined in the site map. The site map has three navigation components:

- Areas (1):
- Groups (2)
- Pages (3)

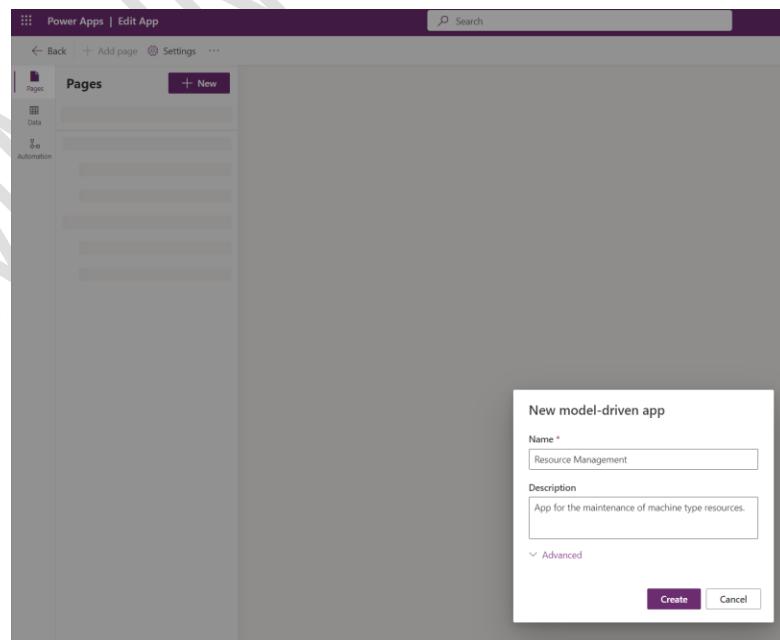


Knowing the basics, we will start creating the model-driven app:

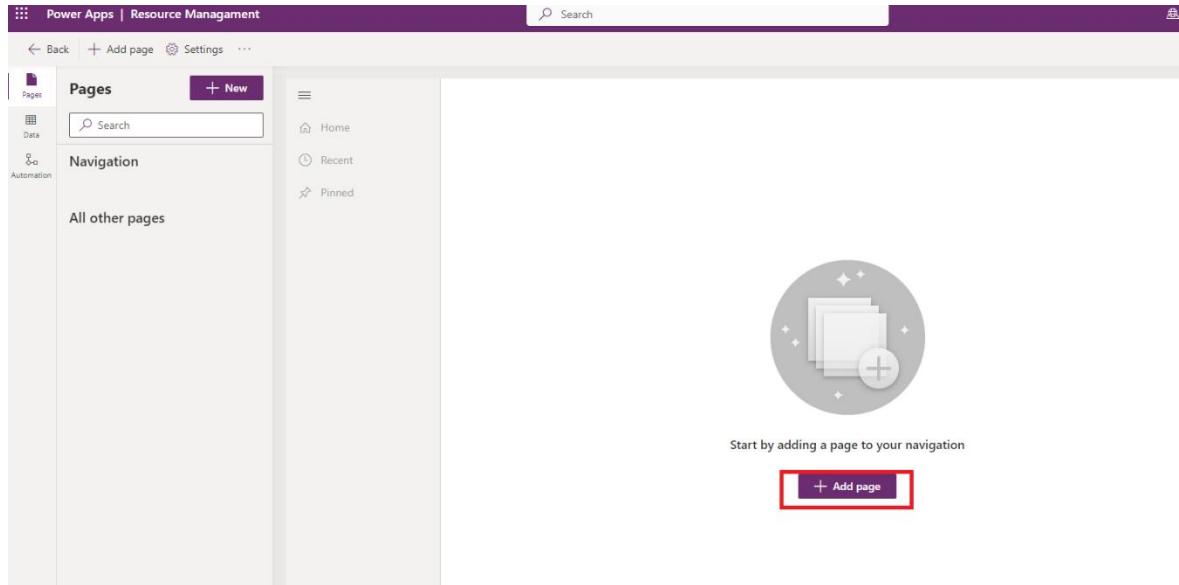
1. Open the solution, and click on New -> App -> Model-driven app.



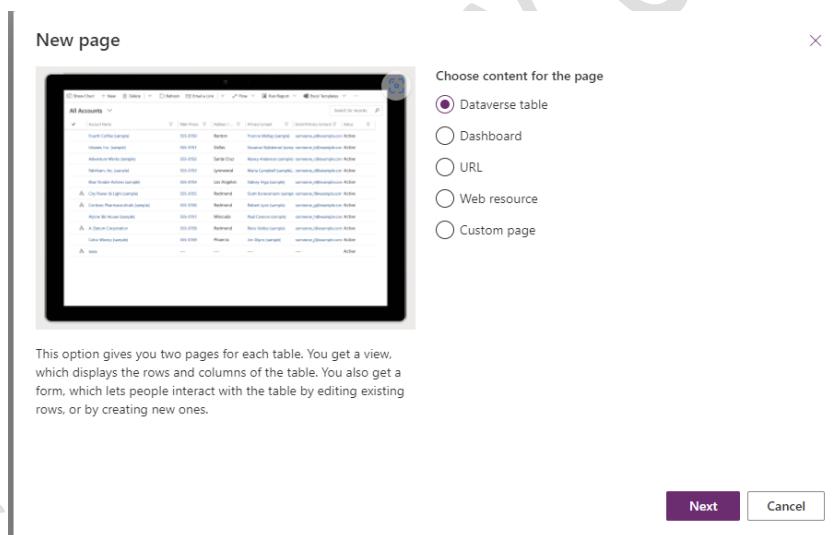
2. The model-driven app editor will open. The first thing to do is naming the app and click on create.



- The next step is to select the tables that we want to add to our model-driven app: click on add page.



- Select “Dataverse table” and click on Next.



- Select the Resource, Maintenance Person and MRO Service tables and click on Add.

**New page**

Choose data table for these pages

Select existing table

Create new table

Select one or more tables

Resource

Ms Graph Resource To Subscription

Resource

Show in navigation

Add Cancel

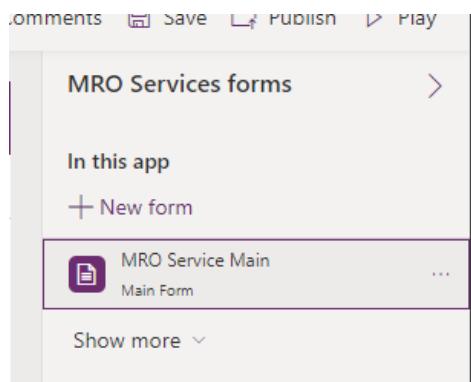
This option gives you two pages for each table. You get a view, which displays the rows and columns of the table. You also get a form, which lets people interact with the table by editing existing rows, or by creating new ones.

## 6. The tables are added as pages to our app...and we are almost ready!

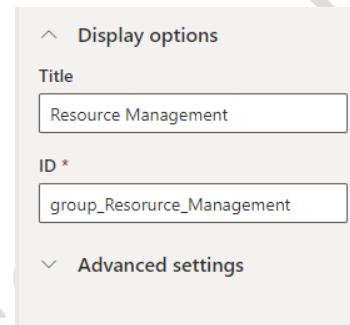
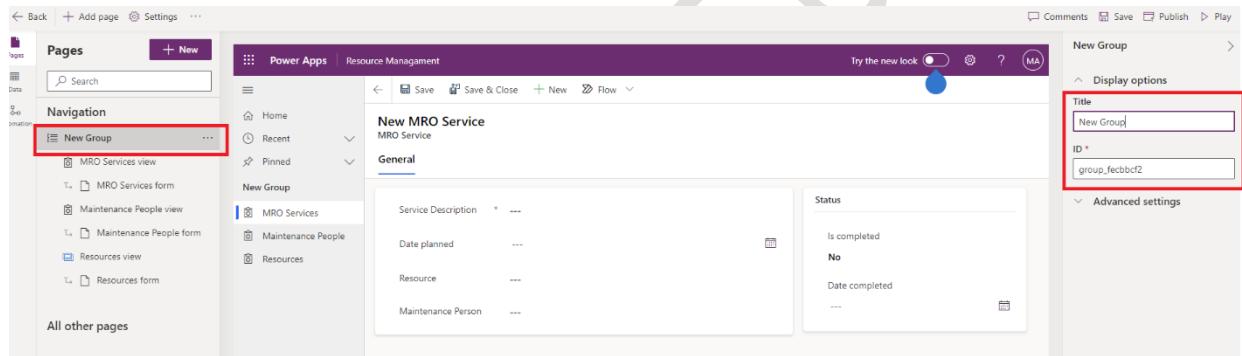
Notice that for every page shown in the navigation menu we have views and forms available.

Check that every element in the navigation menu contains the views and pages that we design in previous step.

You can also add more forms or views to every page if they are needed:



7. Click in “New Group” in the navigation pane and change the title for “Resource Management” and the ID for “group\_Resource\_Management.”

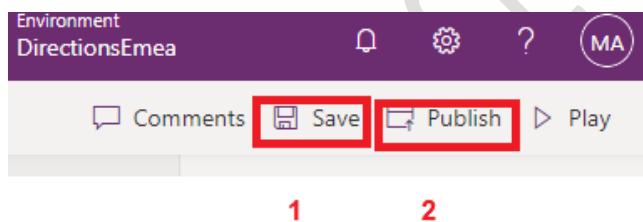


8. As an optional step, you can check every page to change the name to a more friendly term and change the associated icon:

The screenshot shows the Power Apps interface for 'Resource Management'. On the left, there's a navigation pane with 'Pages' and 'MRO Services view' highlighted. The main area displays a table titled 'Active MRO Services' with one record: 'R0010 - SL...' for 'Regular Maintenance'. On the right, the 'Settings' tab is selected in the ribbon, showing details like 'Content type: Table', 'Title: MRO Services', and 'Icon: Default icon'. A red box highlights the 'Icon' section.

The icons are web resources inside your solution. You can add new icons if needed, as is recommended that every table have an icon associated with to a better user experience. You can learn more about icons [here](#).

9. A first version of the app is ready! Click on Save ant then Publish!



10. Execute the app clicking on Play and add some new records to your new app!

Create maintenance persons, complete the information of the resources, and insert planned MRO Services!

Hint: You can try the latest look of model-drive apps!

A screenshot of the latest look of model-drive apps. It shows a table with columns: 'Completed' (sorted), 'Date completed', and 'Created On'. The 'Created On' column shows a value of '10/31/2023 12:50 AM'. The top bar includes a 'Try the new look' toggle, search, filter, settings, and share icons. Below the table are 'Edit columns', 'Edit filters', and 'Filter by keyword' buttons.

### **Adding Security (optional)**

Once our app is ready, we need to grant access to the users. The model-driven app security is role based. If the app contains custom tables, as is our case, and user with power platform admin roles need to configure privileges on the new tables in a [security role](#).

More information regarding security [here](#).

COMMUNITY USE ONLY!

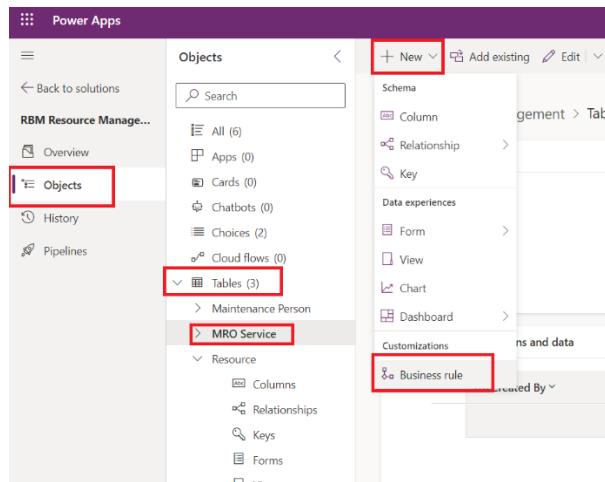
## Extra Step – Developing a Business Rule (optional)

Imagine that you want to show the “Date Completed” field on the “MRO Services” forms only when the service is marked as complete. This can be easily solved using a business rule.

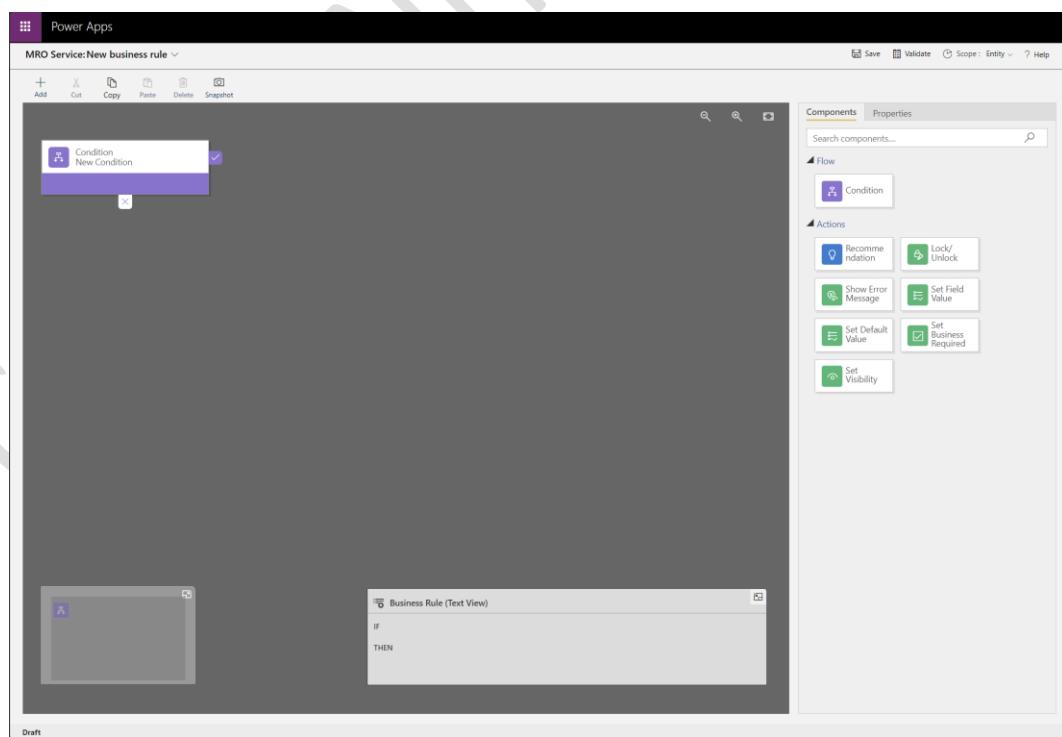
A [business rule](#) is defined in the table and allows us to implement logic without writing any code. The scope can be the table, some or all forms.

Let us start with our Business Rule:

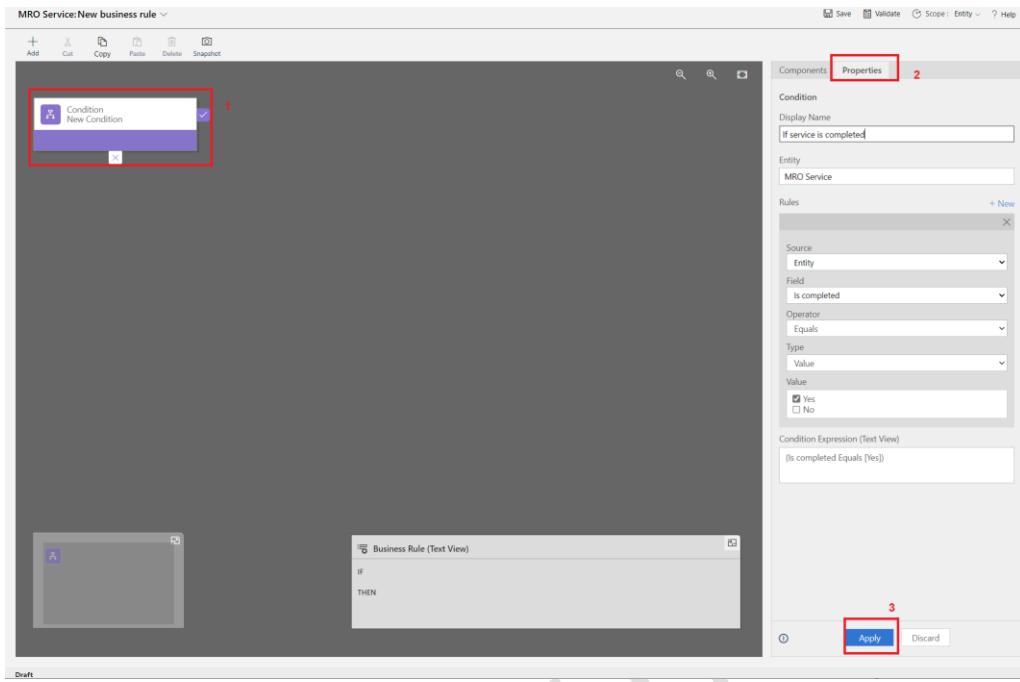
1. Select the MRO Service Table and click on New -> Customization -> Business Rule



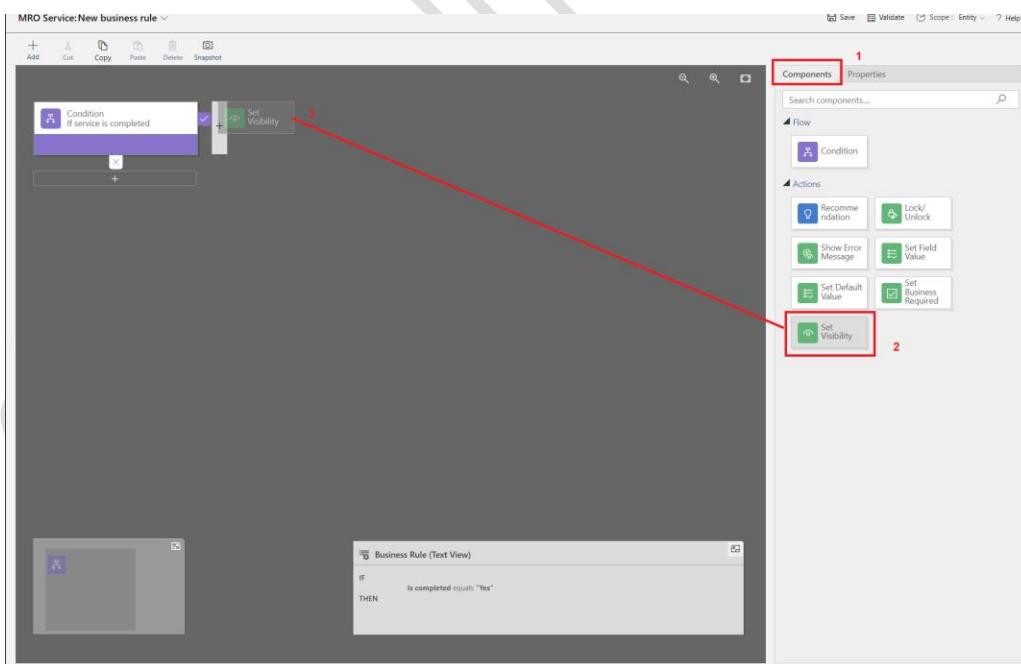
2. The business rule designer will open in a new tab:



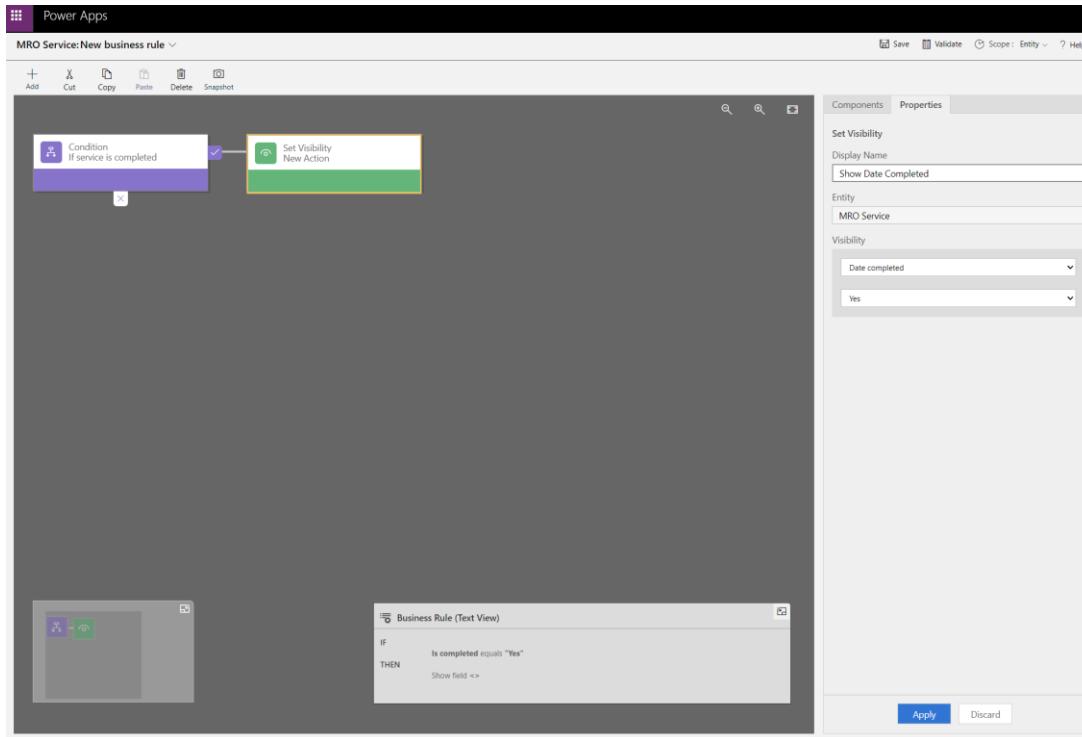
3. Click on the “Condition” box, select Properties, set the condition as show in the screenshot and click “Apply”:



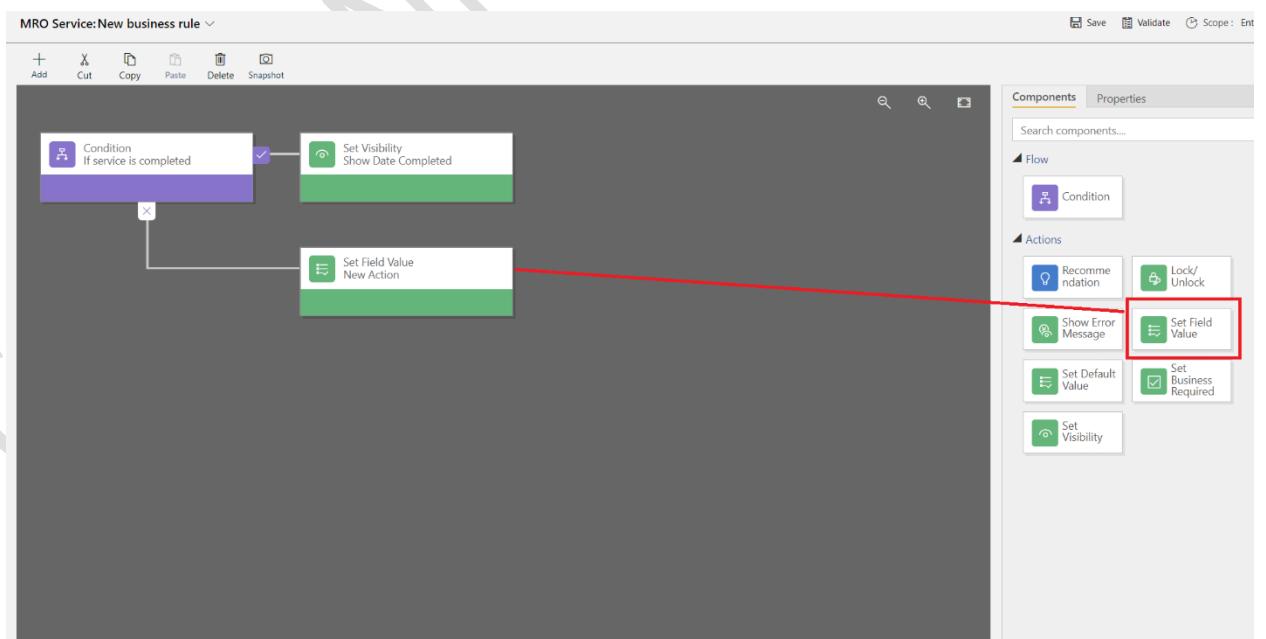
4. Click on components, select “Set visibility”, and drag and drop it to the canvas connecting it to the condition:



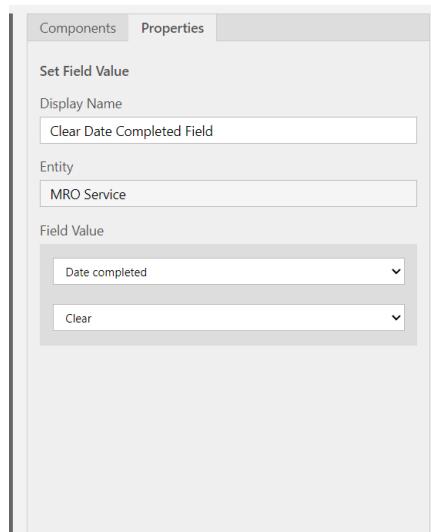
5. Click on the added box and fill the properties as shown in the screenshot, and click on apply:



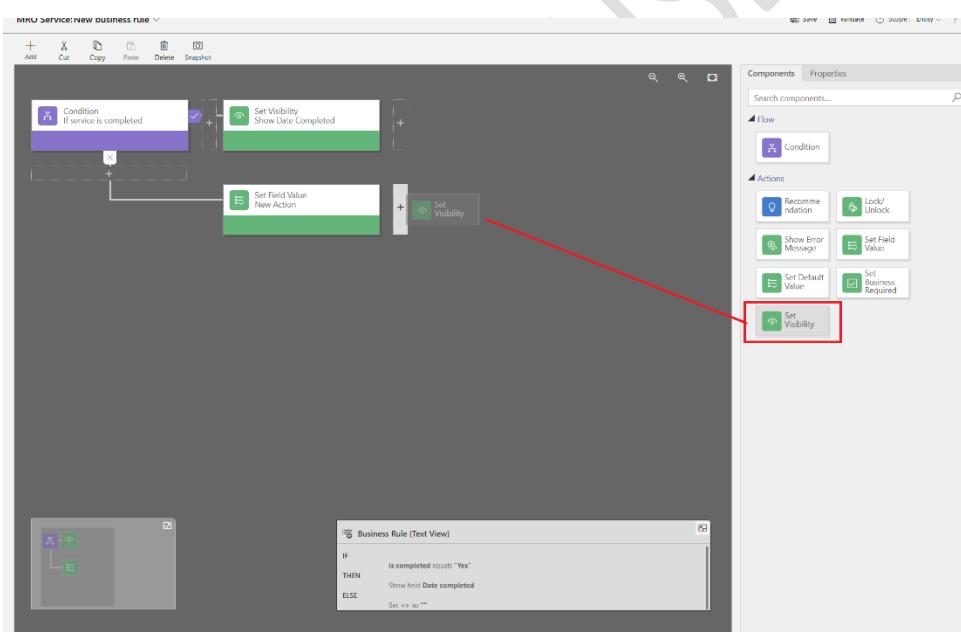
6. If the “Is completed” field, is changed to “No”, we need to clear the field and make it no visible again. Select the “Set Field Value” component.



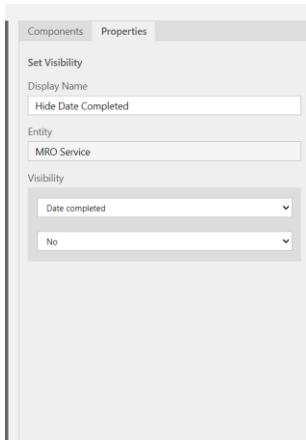
And fill the properties as shown in the next screenshot and click on “Apply.”



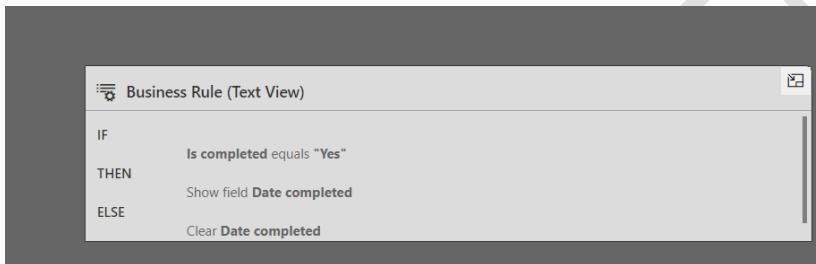
7. Select the component “Set visibility” and drop it as the next step on the “false” branch:



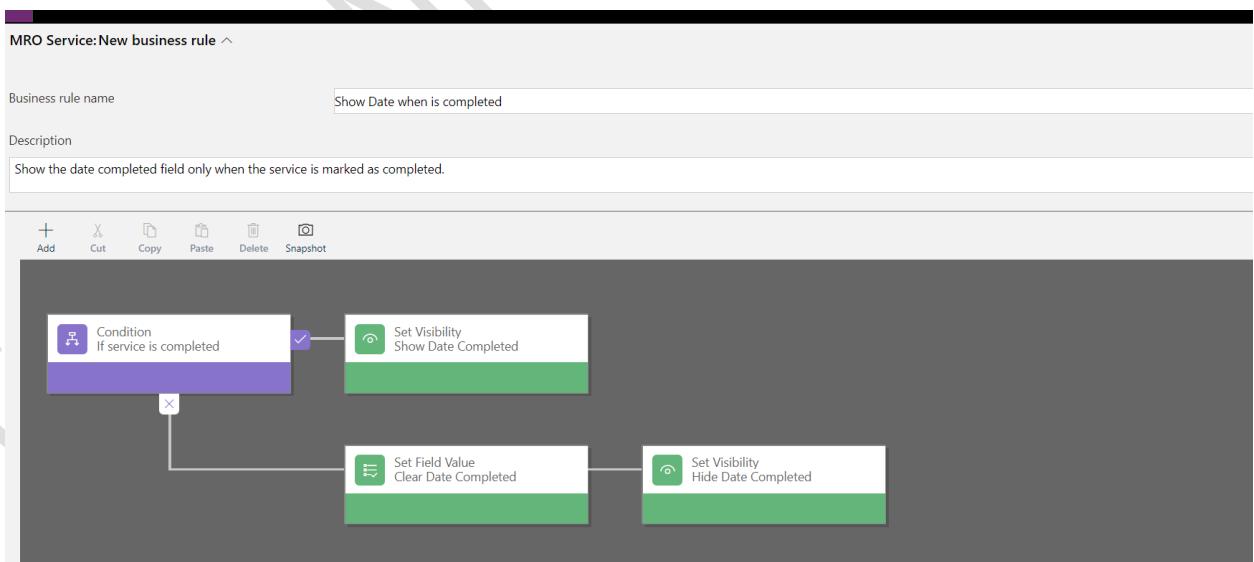
Just set up the properties as shown and click on “Apply”:



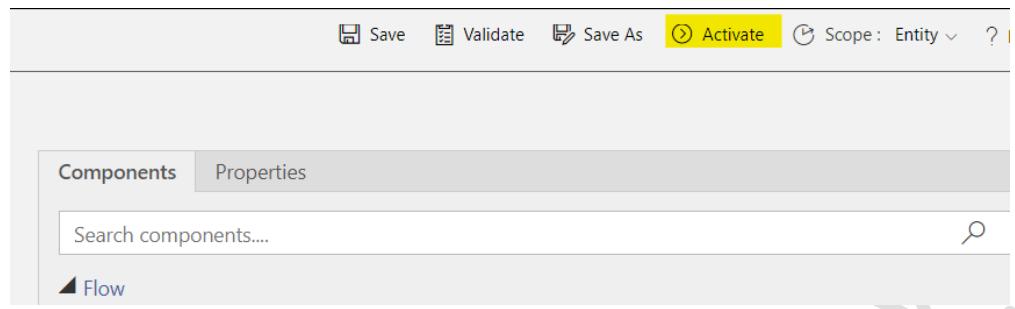
8. Now you can look at the Text View of the Business Rule and check if everything is ok.



9. And before saving add a name and a description to the Business Rule:



10. After saving, the Business Rule needs to be activated to apply:



The rule can also be turned on from the solution designer:

A screenshot of the Microsoft Dynamics 365 Business Central solution designer. On the left, the 'Objects' tree shows 'Tables' expanded, with 'MRO Service' selected. In the main area, a list of business rules is shown, with one rule named 'Show Date when is completed' selected. A context menu is open over this rule, with 'Turn on' highlighted in grey. Other options in the menu include 'Edit', 'Advanced', and 'Remove'. The status of the rule is listed as 'Off'.

## Extra Step – Try the new formula type column.

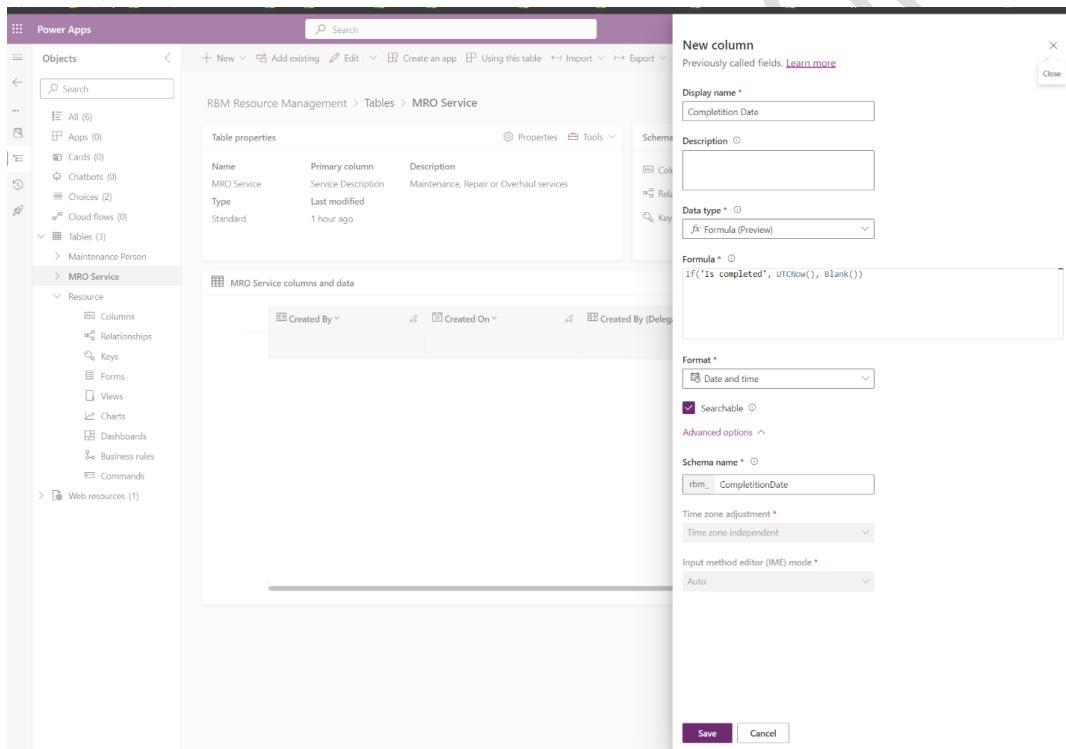
Users give us feedback and ask if it is possible to auto populate the date when the service is done when the select “Yes” in the “Is complete” field.

You can solve that, in a low code approach, using the new formula type columns.

<https://learn.microsoft.com/en-us/power-apps/maker/data-platform/formula-columns>

To achieve that, try to add a new column on the MRO Services table, select the Formula data Type and enter this formula:

If('Is completed', UTCNow(), Blank())



Place the new column on the main form of MRO Services table to check how it works! If everything is ok, you can delete the old column that show the date when the service is complete.