

WORKSHOP: Model-Driven Apps for beginners

RAUL BOGAJO

WORKSHOP: MODEL-DRIVEN APPS FOR BEGINNERS

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 create a simple (but cool...) model-Driven App.

Scenario

CRONUS enterprise works with machine type resources and need to keep track of the maintenance of them. At this moment they only want to register the date of maintenance, the person who performs the maintenance task 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.

COMMUNITY USE ONLY!

Step 1 - Creating a solution.

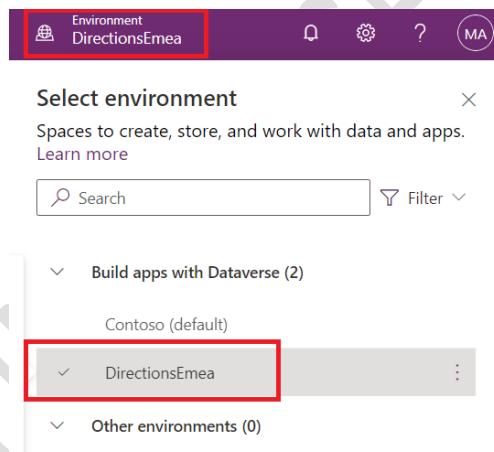
A Dataverse Solution is a way to package the different components that are part of a software unit developed with the Power Platform. Solutions are also used to move the packages between environments and to keep track about the authoring.

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

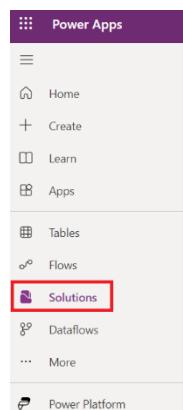
It is always a good practice, for the reasons mentioned above, to keep all the things together in a solution.

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

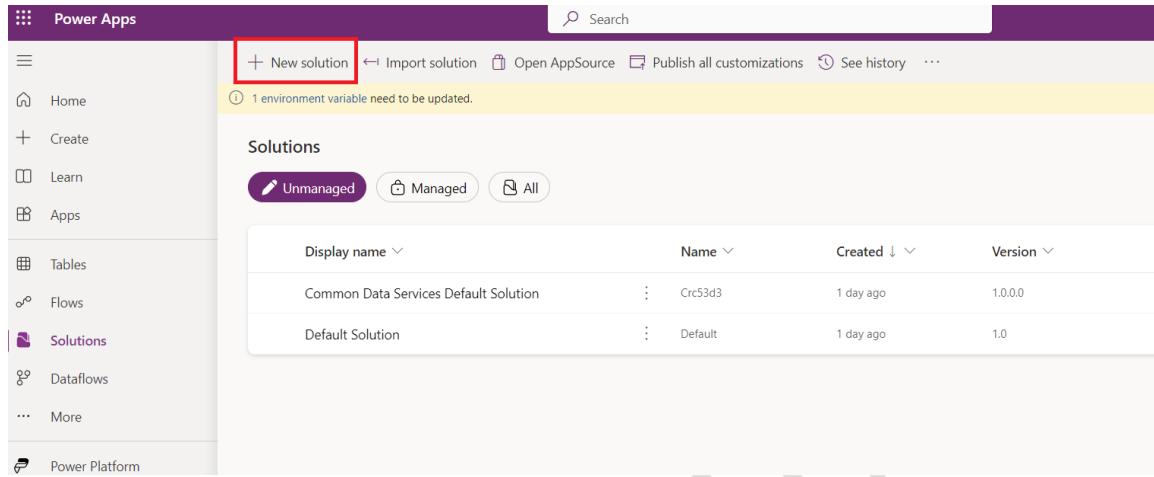
1. Navigate to [make.powerapps.com](#) and log in using your assigned credentials.
2. Select, on the top right corner, the [environment](#) where you are going to create the solution
(HINT: Avoid the deploying solutions in the default environment)



3. Select Solutions on the left pane.



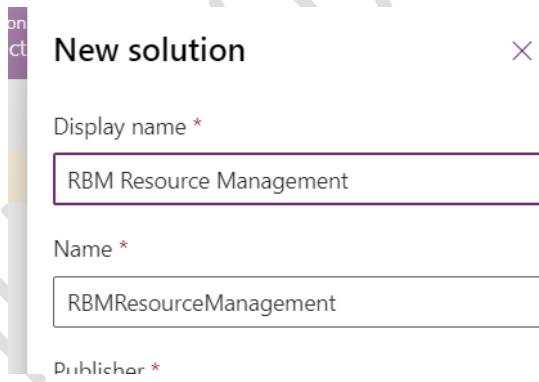
4. Click on new solution:



The screenshot shows the 'Solutions' section of the Power Apps portal. On the left, there's a navigation menu with options like Home, Create, Learn, Apps, Tables, Flows, Solutions (which is selected and highlighted in purple), Dataflows, More, and Power Platform. At the top right, there are buttons for '+ New solution', 'Import solution', 'Open AppSource', 'Publish all customizations', 'See history', and more. A yellow banner at the top indicates '1 environment variable need to be updated.' Below the banner, the 'Solutions' table lists two entries: 'Common Data Services Default Solution' and 'Default Solution'. The 'Display name' column shows the names, 'Name' shows the internal IDs, 'Created' shows the time, and 'Version' shows the version number.

Display name	Name	Created	Version
Common Data Services Default Solution	Crc53d3	1 day ago	1.0.0.0
Default Solution	Default	1 day ago	1.0

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 three fields: 'Display name *' containing 'RBM Resource Management', 'Name *' containing 'RBMRessourceManagement', and 'Publisher *' which is currently empty. There are also 'Save' and 'Cancel' buttons at the bottom.

New solution

Display name *

RBM Resource Management

Name *

RBMRessourceManagement

Publisher *

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 *

Name *

Publisher * (CDS Default Publisher (Crff3ae))

Raul Bogajo (RaulBogajo)

More options ^

Installed on (Calendar icon)

- 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 *

Name *

Publisher * (CDS Default Publisher (Crff3ae))

+ New publisher

Version *

More options ^

Installed on (Calendar icon)

Configuration page

Description

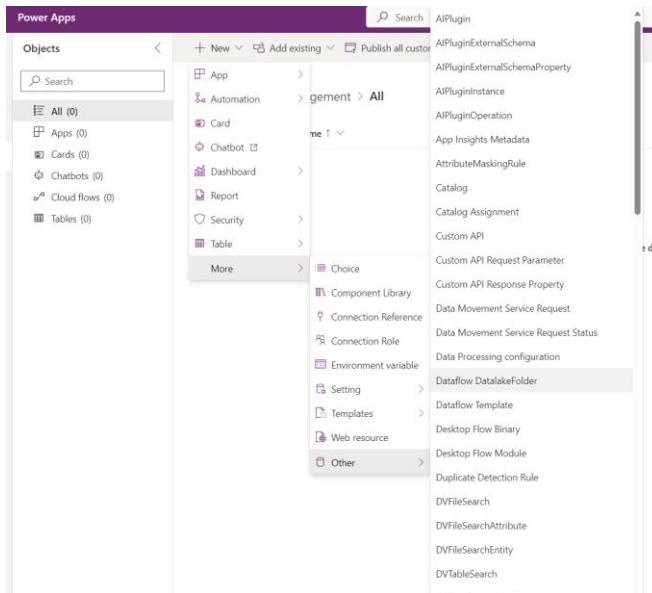
Package type (Unmanaged)

Create Cancel

- And the solution is ready! The next step is adding the components!

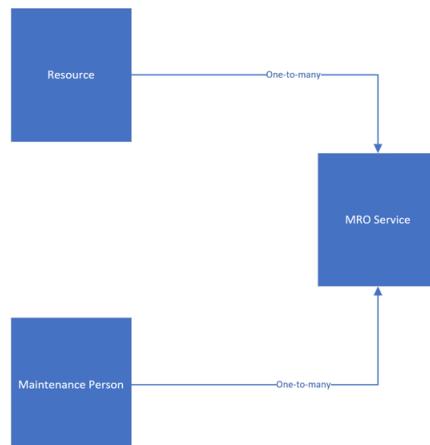
Step 2 - Creating the data model: tables and relationships.

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 red box highlights the 'Edit' button next to the 'RBM Resource Management' solution. The table lists three solutions: 'RBM Resource Management' (selected), 'Common Data Services Default Solution', and 'Default Solution'. The 'RBM Resource Management' row has columns for Display name, Name, Created, and Version.

2. Select New -> Table -> Table.

The screenshot shows the 'Power Apps' interface with the 'Objects' section selected. A red box highlights the 'New' dropdown menu. The menu is expanded to show options: 'App', 'Automation', 'Card', 'Chatbot', 'Dashboard', 'Report', 'Security', 'Table' (which is highlighted with a red box), and 'More'. Below the menu, it says 'Table from external data'.

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 *
Resource

Plural name *
Resources

Description
Machine type resources imported from Business Central

Enable attachments (including notes and files) ¹

[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 *
Name

Description
Resource name

[Advanced options](#) ▾

Schema name *
rbm__ Name

Column requirement *
Business required

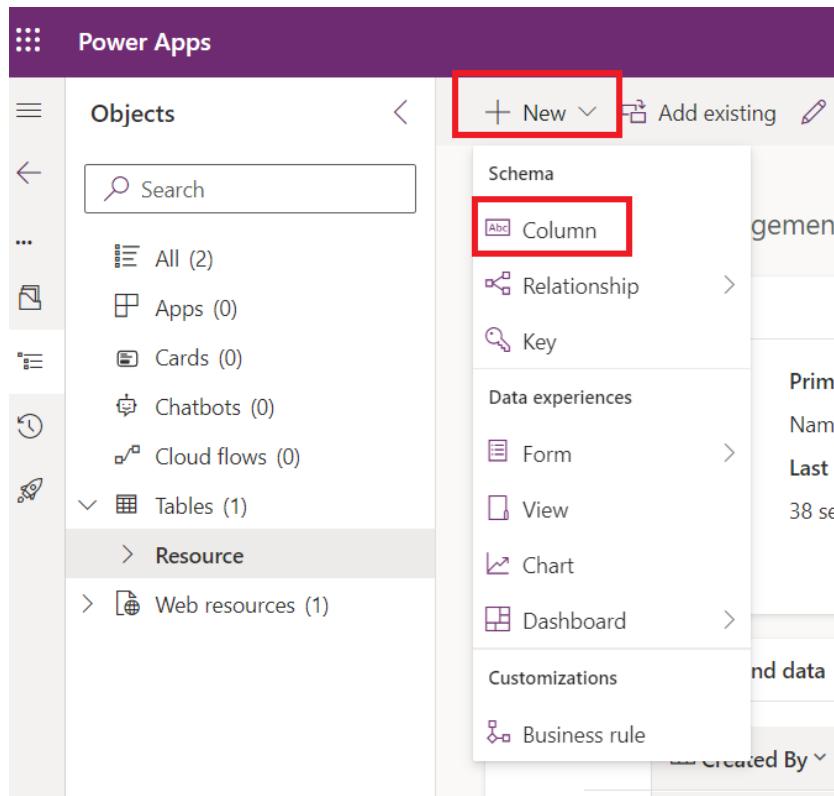
Maximum character count *
100

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 *

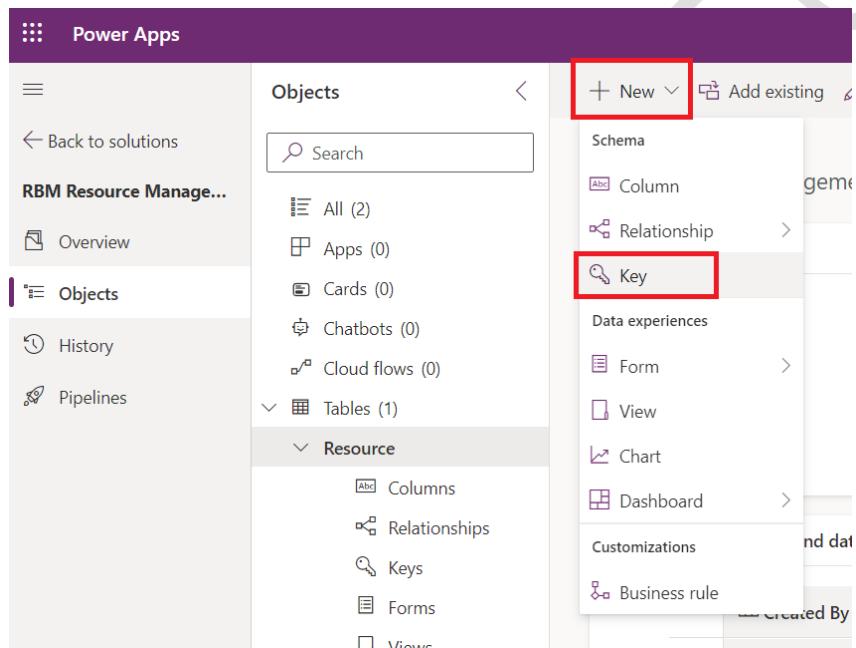
- 8. If everything is ok (looking as in the previous screenshot), click on save, and the column is ready!

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

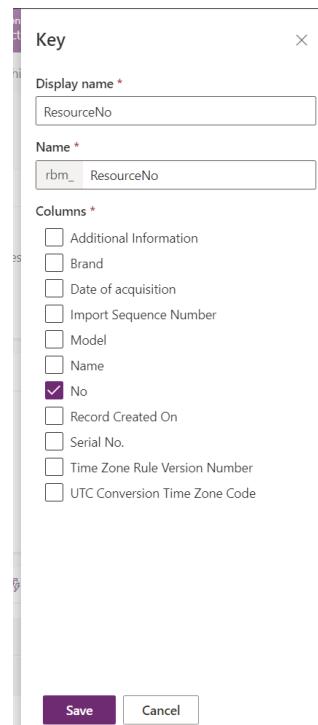
Field Name	Data Type	Format	Length	Comment
Brand	Text	Text	100	Commercial brand of the resource
Model	Text	Text	100	Model of the resource
Date of acquisition	Date and Time	Date Only		Date when the resource was acquired.
Serial No.	Text	Text	100	Serial no. of the resource
Additional Information	Text	Text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

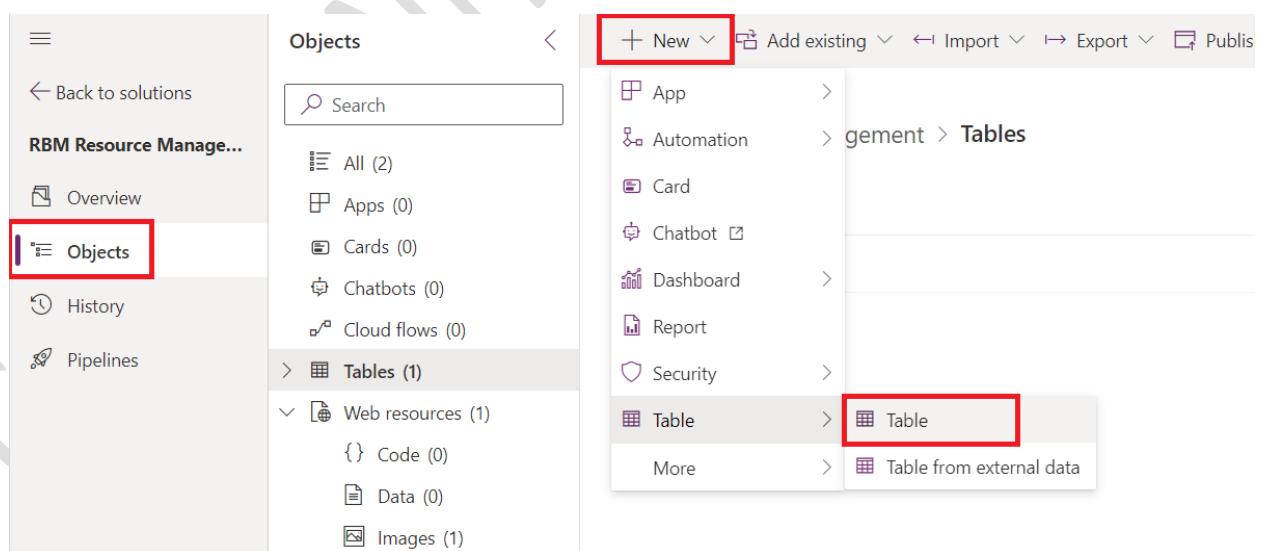
To create the key, click on New -> Key.



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



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



Maintenance Person:

Display Name	Data Type	Format	Length	Comment
Name	Text	Text	100	Primary Column
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

Choices	Sort
Label * <input checked="" type="checkbox"/> Technician	Value * 909,250,000
Label * <input checked="" type="checkbox"/> Supervisor	Value * 909,250,001
Label * <input checked="" type="checkbox"/> Manager	Value * 909,250,002

+ New choice

Advanced options

MRO Service:

Display Name	Data Type	Format	Length	Comment
Service Description	Text	Text	100	Primary Column
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.

Display name *

Choices

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

+ New choice

Advanced options ▾

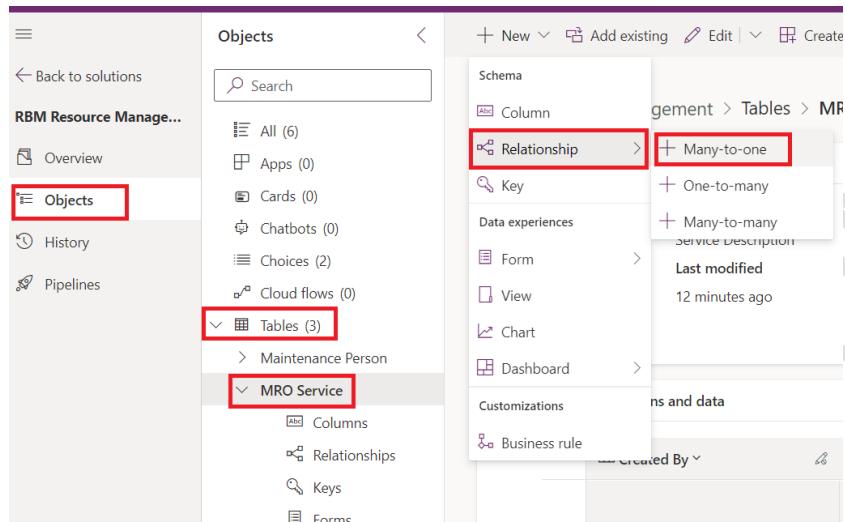
Notice that both new choices are created as objects, and added to the solution:

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

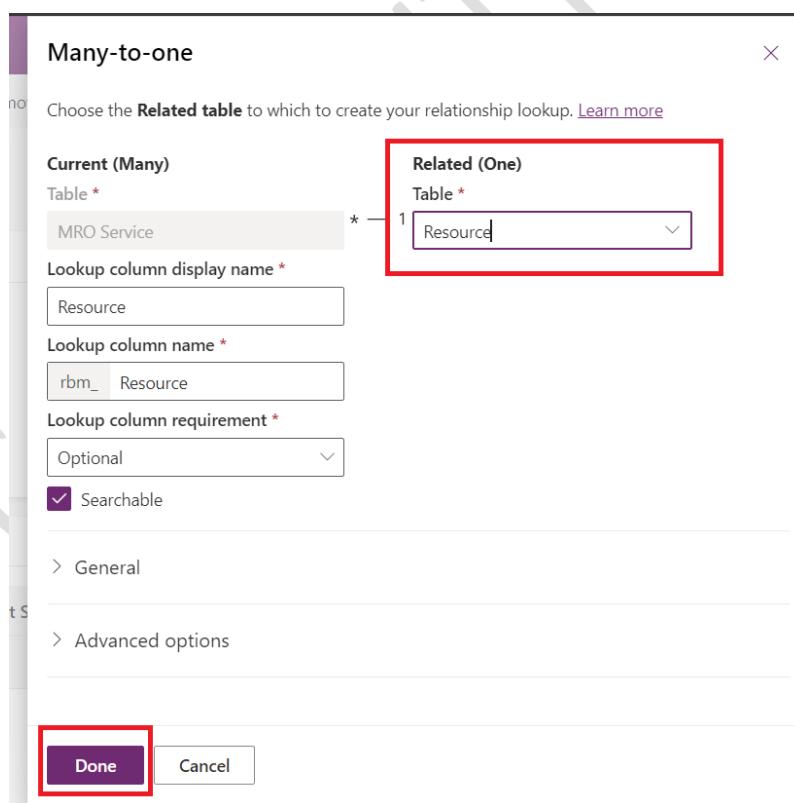
13. After all tables are created, we are going to establish the [relationships](#) between them.

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 portal interface. On the left, there's a navigation sidebar with 'Objects' selected, showing categories like All, Apps, Cards, Chatbots, Choices, Cloud flows, Tables (3), Maintenance Person, MRO Service, Resource, Columns, Relationships, Keys, Forms, Views, Charts, Dashboards, Business rules, Commands, and Web resources. Under 'Relationships', the 'Relationships' section is highlighted. In the center, the 'RBM Resource Management > Tables > Resource > Relationships' page is displayed. It lists various relationships with columns for Display name, Name, Related table, Relationship type, Managed, and Customizable. One relationship, 'Resource', is highlighted in yellow and has its details shown in a preview pane: Name: rbm_MROService_rbm_Resource_rbm_Resource, Related table: MRO Service, Relationship type: One-to-many. The preview pane also shows the 'rbm_MROService_rbm_Resource_rbm_Resource' column with a dropdown icon.

Step 3- 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. The 'Table properties' card shows the name 'Maintenance Person', primary column 'Name', and description 'List of the maintenance staff'. The 'Schema' card lists 'Columns', 'Relationships', and 'Keys'. The 'Data experiences' card shows 'Forms' (with 'Views' highlighted), 'Charts', and 'Dashboards'. The 'Customizations' card lists 'Business rules' and 'Commands'. At the bottom, there's a grid of columns labeled 'Created By', 'Created On', 'Created By (Delegate)', 'Import Se...', 'Modified By', and '+16 more'. A red box highlights the 'Views' link under the 'Forms' section in the 'Data experiences' card.

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

The screenshot shows the Power Apps portal interface. On the left, there's a sidebar with navigation links: 'Back to solutions', 'RBM Resource Management...', 'Overview', 'Objects' (which is selected and highlighted in grey), 'History', and 'Pipelines'. The main area is titled 'Objects' and shows a list of items under '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 'Edit' highlighted by a red box in the top right corner. The views include: 'Active Maintenance People' (selected and highlighted with a grey box), 'Inactive Maintenance People', 'Maintenance Person Advanced Find View', 'Maintenance Person Associated View', 'Maintenance Person Lookup View', and 'Quick Find Active Maintenance People'.

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 'Table columns' editor for the 'Maintenance Person' table. On the left, a list of columns includes 'Charge' (which is highlighted with a red box). On the right, the 'View' pane shows the current column order: 'Name ↑' and 'Created On ↓'. A red arrow points from the 'Charge' column in the list to its position in the view header, indicating where it should be dragged. Below the view, a message says 'We didn't find anything to show here'.

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

The screenshot shows the Microsoft Dynamics 365 Model-driven App builder interface. On the left, there's a sidebar titled "Table columns" for the "Maintenance Person" entity. It lists various system fields like Created By, Modified By, and Status. The main area displays a view titled "Active Maintenance People" with columns for Name, Charge, Email, Phone No., and Created On. A message at the bottom of the view area says "We didn't find anything to show here" and "This view does not have any data. Try creating more records or removing filters." There's also a circular icon with a grid symbol.

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

The screenshot shows the Microsoft Dynamics 365 ribbon. The "Save and publish" button is highlighted in green, indicating it has been selected. The ribbon also includes other buttons for Validate, Save As, and a dropdown menu.

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

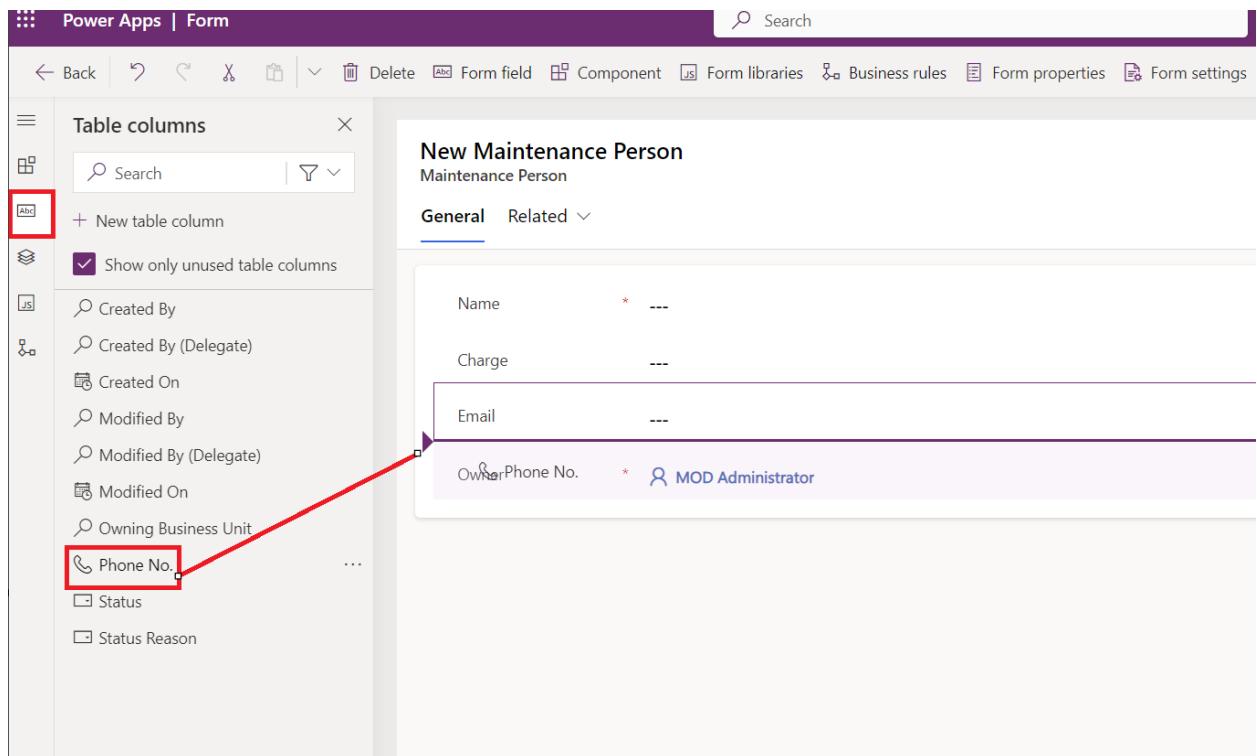
7. Confirm that you still have selected the “Maintenance Person” table and click on Forms.

The screenshot shows the Power Apps portal interface. On the left, the navigation pane includes 'Back to solutions', 'RBM Resource Management...', 'Overview', 'History', 'Pipelines', and a 'Objects' section with 'Tables (3)' expanded, showing 'Maintenance Person' selected. The main area displays 'RBM Resource Management > Tables > Maintenance Person'. The 'Table properties' section shows Name: Maintenance Person, Primary column: Name, Description: List of the maintenance staff, Type: Standard, and Last modified: 21 hours ago. To the right, there are sections for 'Schema' and 'Data experiences'. Under 'Data experiences', the 'Forms' option is highlighted with a red box. Below the table properties, a grid titled 'Maintenance Person columns and data' lists columns: Created By, Created On, Created By (Delegate), Import Se..., and Modified By.

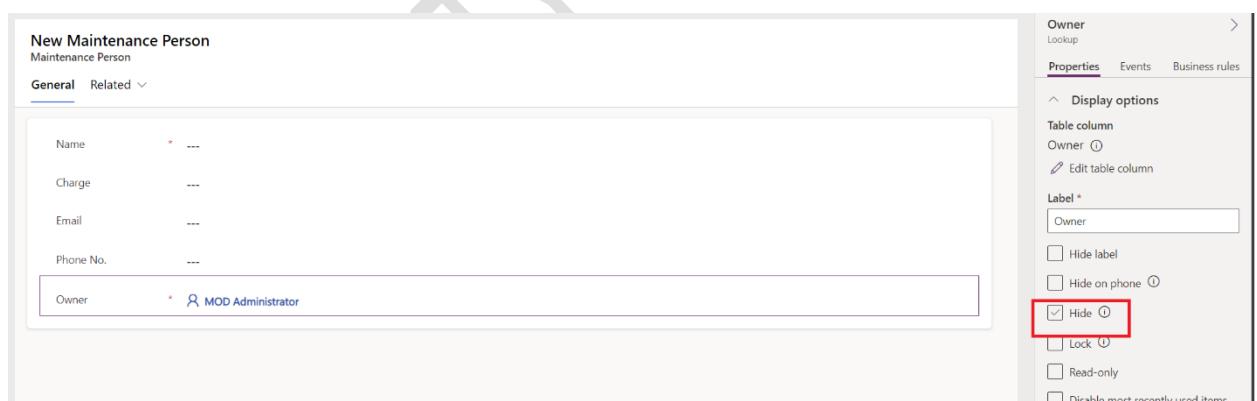
8. The Model-driven apps have [different form types](#). We will edit the main form. Select it and click on edit.

The screenshot shows the Power Apps portal interface. The 'Forms' section in the navigation pane is selected. The main area displays 'RBM Resource Management > Tables > Maintenance Person > Forms'. A table lists three forms: 'Information' (Quick View, Card, Main), 'Information' (Card), and 'Information' (Main). The 'Edit' button in the top toolbar is highlighted with a red box. The 'Information' row with the Main form type is also highlighted with a red box.

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 builder 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'. A checked checkbox 'Show only unused table columns' is visible. The main area displays a form titled 'New Maintenance Person' for the 'Maintenance Person' entity. It has two tabs: 'General' (selected) and 'Related'. The 'General' tab contains fields for 'Name', '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”.

This screenshot shows the 'Form properties' pane open on the right side of the Power Apps Form builder. The 'Display Name' field is set to 'Maintenance Person Main Form' and the 'Description' field is set to 'Main form for maintenance person table.' The 'Max Width (pixels)' field is set to 1920 and the 'Show image' checkbox is checked. The 'Save and publish' button at the top right is highlighted with a red box.

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, and the 'Views' option is highlighted with a red box. Below this, there's a grid for 'MRO Service columns and data' with various columns listed.

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 'Views' option in the sidebar is highlighted with a red box. The main table lists several 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'.

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.

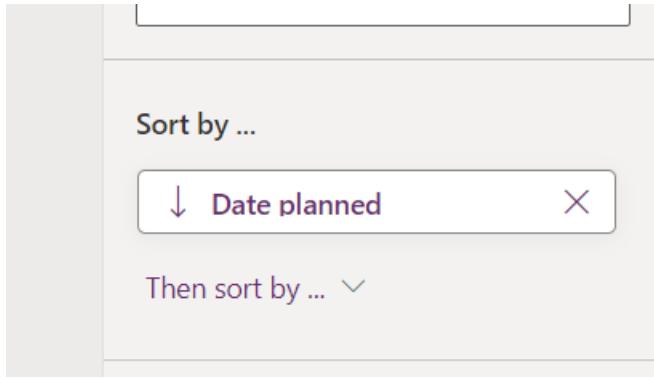
The result view should look like the screenshot below:

The 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 up), 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.' On the left, a sidebar titled 'Table columns' lists fields for 'MRO Service' and 'Related'. Fields listed include: Created By, Created By (Delegate), Maintenance Person, Modified By, Modified By (Delegate), Modified On, Owner, Owning Business Unit, Record Created On, Service Type, Status, and Status Reason.

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

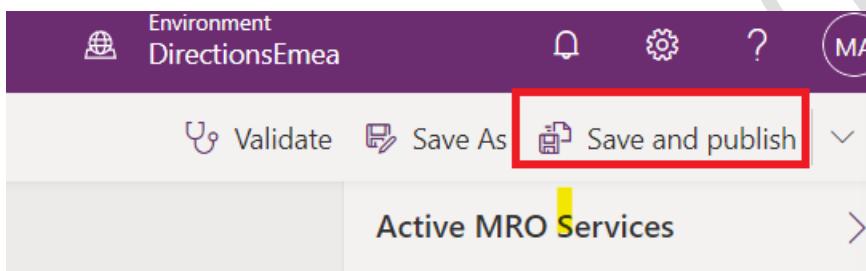
The screenshot shows the Power Apps View interface. The top navigation bar includes 'Search', 'Environment Directions Emea', and standard application controls. The main area displays a table with the following columns: Service Description (sorted up), Is completed, Date completed, and Created On. Below 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.' On the right, a sidebar titled 'Active MRO Services' shows settings for 'Name' (Active MRO Services) and 'Description'. A 'Sort by ...' section is highlighted with a red border, showing 'Service Description' as the current sort value. A 'Then sort by ...' section is also present. At the bottom, a 'Filter by ...' section shows 'Status is 'Active''. The left pane shows the 'classic' view mode.

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

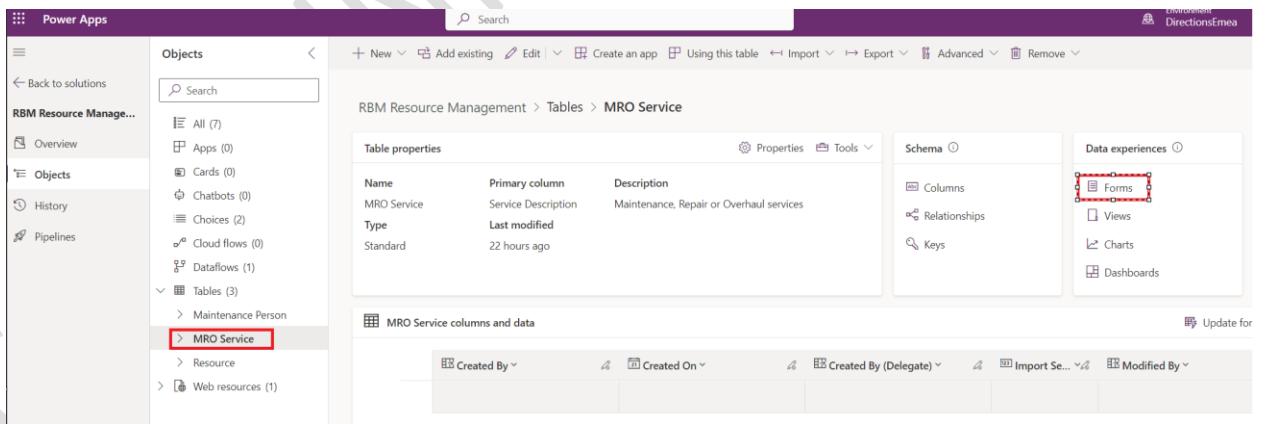


Notice that you can add more columns to define the order of the view.

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



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



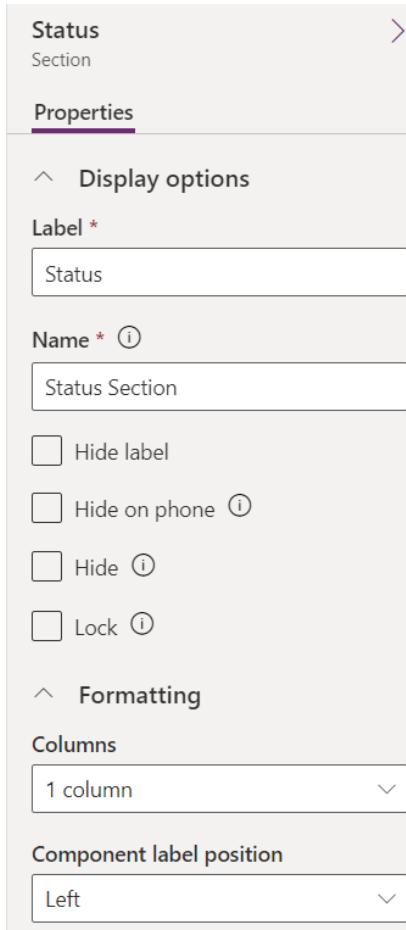
- We are going to customize the main form of this table. Select it and click on “Edit”.

The screenshot shows the Power Apps portal interface. On the left, there's a navigation sidebar with 'RBM Resource Management' selected. Under 'Objects', 'Forms' is highlighted. The main area displays a list of forms under 'MRO Service'. The 'Information' form is listed three times: once as 'Quick View' (Status: On), once as 'Main' (Status: On), and once as 'Card' (Status: On). The 'Main' form is currently selected.

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' form. On the left, the 'Table columns' pane lists various fields like 'Service Description', 'Owner', etc. The main area shows the 'General' tab with some form fields. To the right, the 'Properties' pane is open. A red box highlights the 'Layout' section under 'Formatting'. It shows two options: '1 column' and '2 columns'. The '2 columns' option is selected. A red callout points to this selection with the text '2. change the layout'.

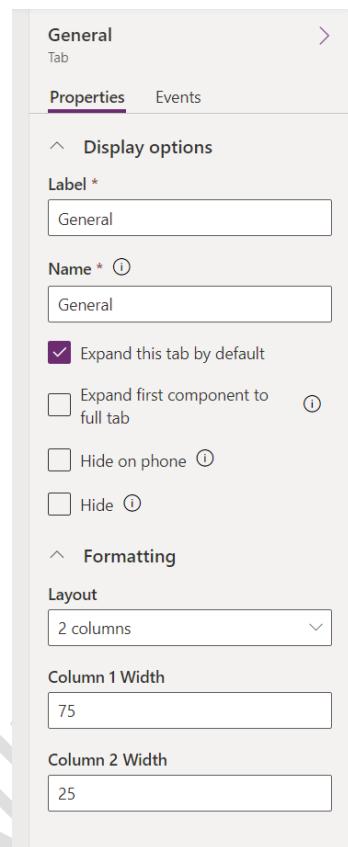
10. A new section will be added to the tab. Define the properties as follows:



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

The screenshot shows a 'New MRO Service' form. On the left, there's a 'General' tab and a 'Related' dropdown. The main area contains fields for 'Service Description' (with a red asterisk), 'Date planned', 'Resource', and 'Maintenance Person', each with a three-dot ellipsis. To the right, there's a 'Status' section with a table. The table has two rows: 'Is completed' (checkbox) and 'Date completed' (date picker). The status is currently set to 'No'.

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' form. The left side displays fields: 'Service Description' (with a red asterisk), 'Date planned', 'Resource', and 'Maintenance Person', each with a three-dot ellipsis button. The right side shows 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 Dynamics 365 Resource Management interface. On the left, there's a sidebar with navigation links like Back to solutions, RBM Resource Management..., Overview, Objects, History, Pipelines, and a expanded section for Tables. Under Tables, the Resource table is selected, and its sub-sections are shown: Columns, Relationships, Keys, Forms, and Views. The Views section is highlighted with a red box. On the right, the main area shows the 'Views' list for the Resource table. The list includes:

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

The 'Active Resources' view is selected and highlighted with a red box. At the top of the page, there are buttons for New view, Add existing view, Edit (which is also highlighted with a red box), Turn off, Set as default view, and Advanced.

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" for the "Resource" table. It lists various columns like "No", "Name", "Brand", etc., along with their descriptions. On the right, the main area displays a table with 10 rows of data, each representing a resource with a unique ID (R0010 to R0100) and a name (e.g., "R0010 - Single Man Lift").

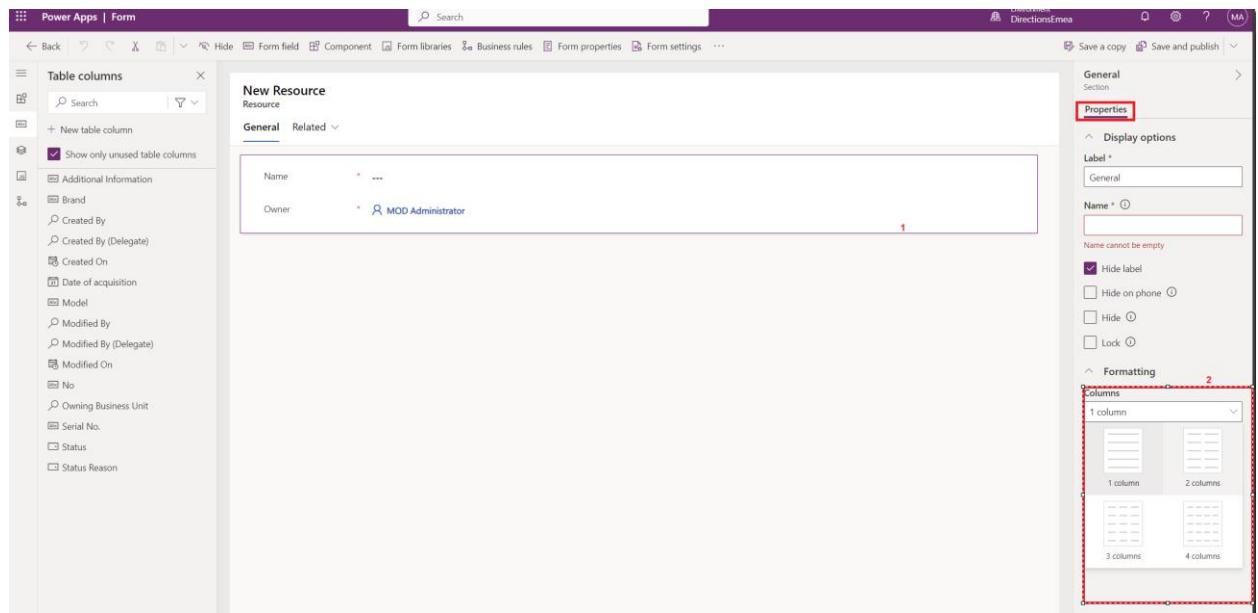
4. When you are ready, save and publish the modified view and go back to the solution.

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

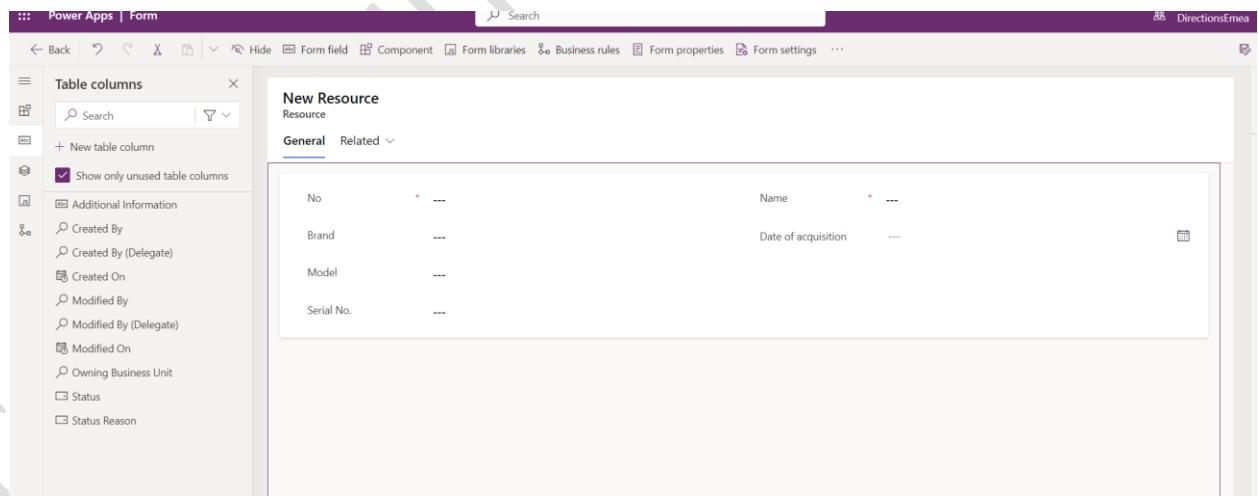
The screenshot shows the Power Apps Objects page. In the left sidebar under "Objects", the "Tables" section is expanded, and the "Resource" table is selected. Below it, the "Forms" option is highlighted with a red box. The main pane shows a list of forms for the Resource table, with one row selected and highlighted with a red box. The selected form is named "Information" and is set as the "Main" form.

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

6. 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:

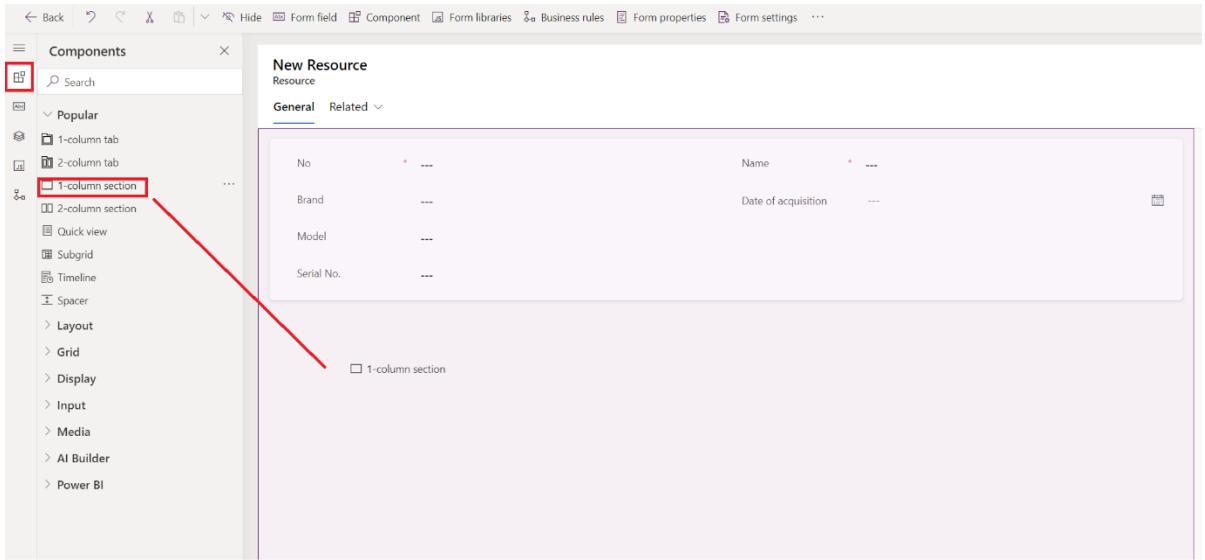


7. Then add the columns to the General section. The result should be something like the next screenshot:

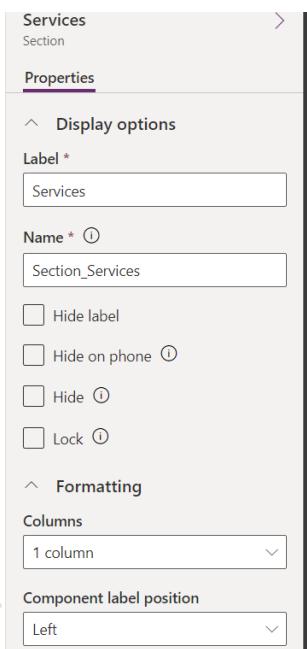


Do not forget to hide the Owner column!

8. 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.

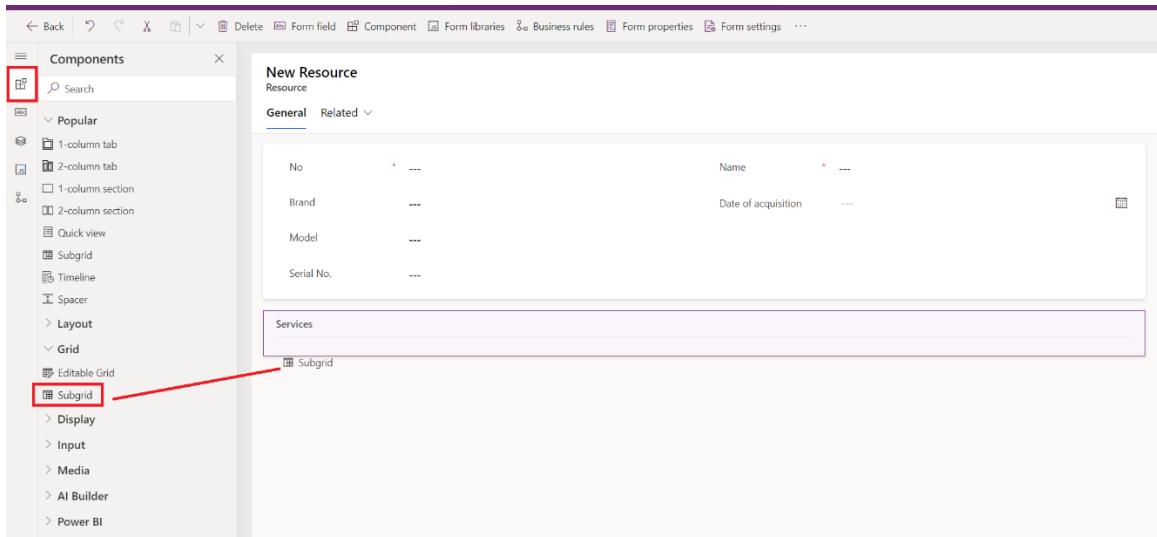


9. Change the value of the Label property to Services and enter Section services as the component name.

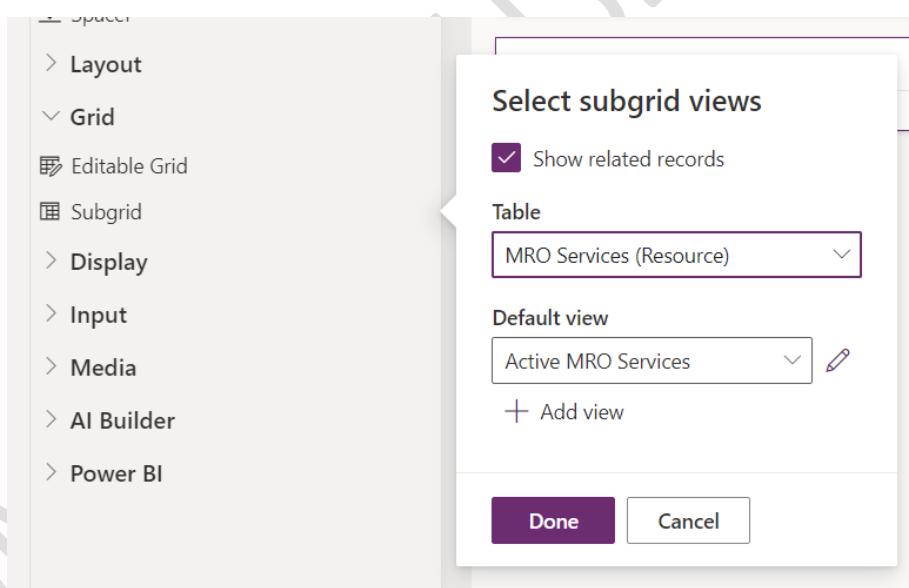


10. We need to show on the form all the services related to the resource displayed. To get this job done, we will add a sub grid to our services section.

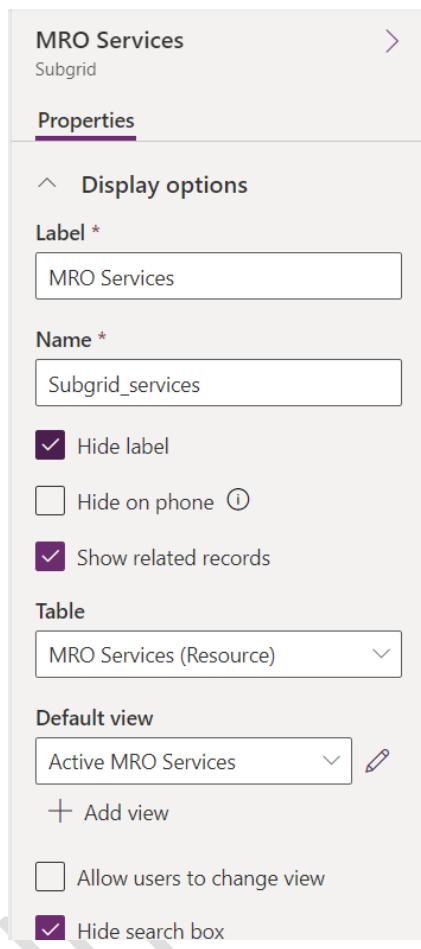
Select the subgrid component and drag it to the brand-new section.



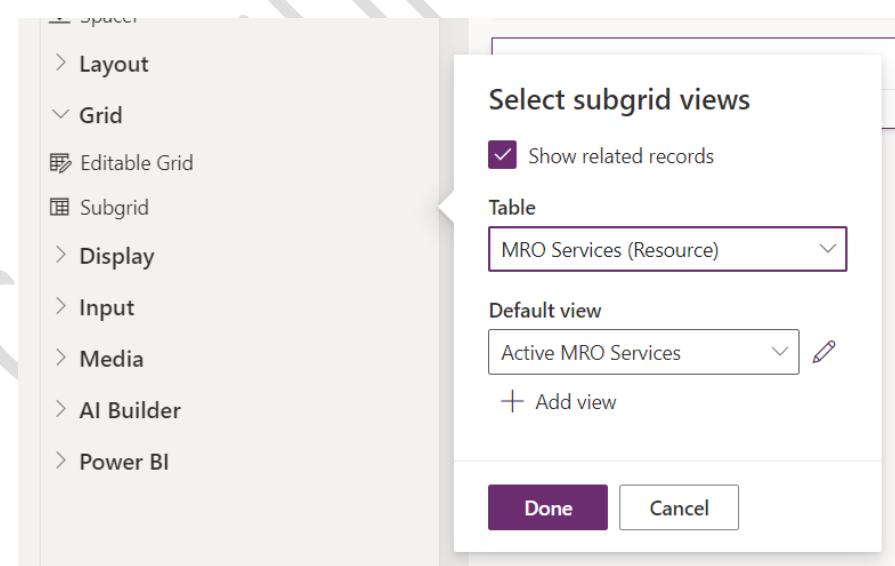
11. As we add the subgrid a pop up will appear to define the subgrid view. Click on “Show related records” to select only related tables to Resource. Select the “MRO Services” table, keep the default view as “Active MRO Services” and click on Done.



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...

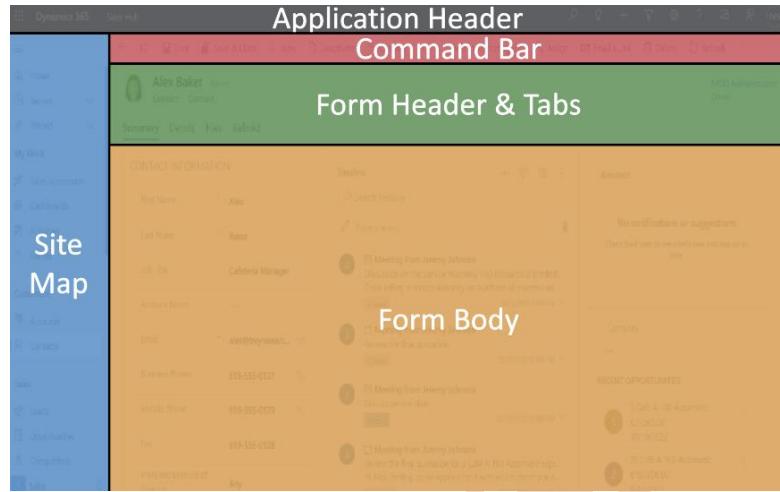


14. ...and Save and Publish!

Shaping the model-driven app

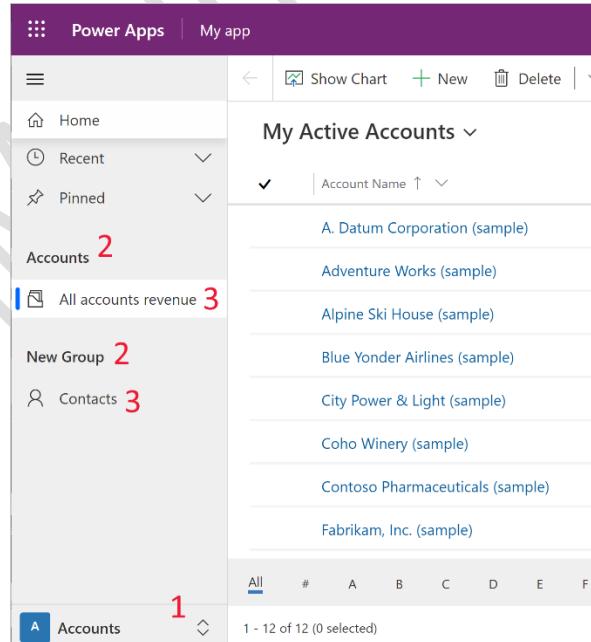
As we defined the views and forms for the table components of our app, the 90% of the work is done and we can create our model-driven app.

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



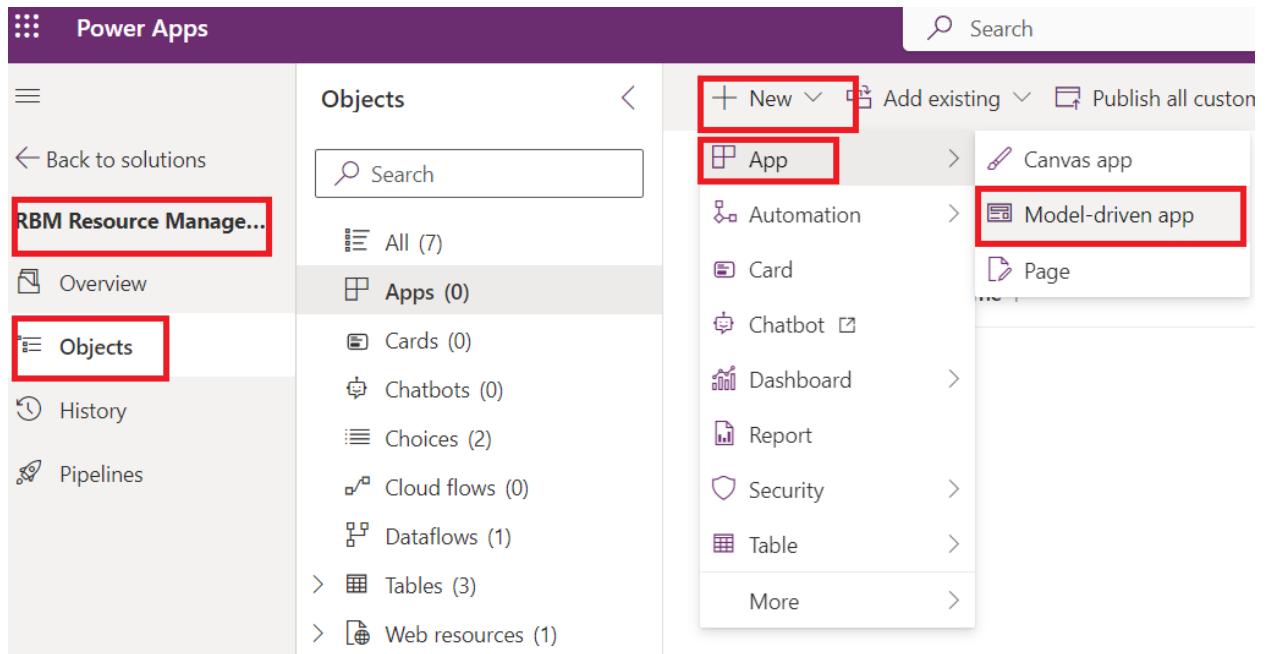
The app navigation is defined in the site map. The site map has three navigation components:

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

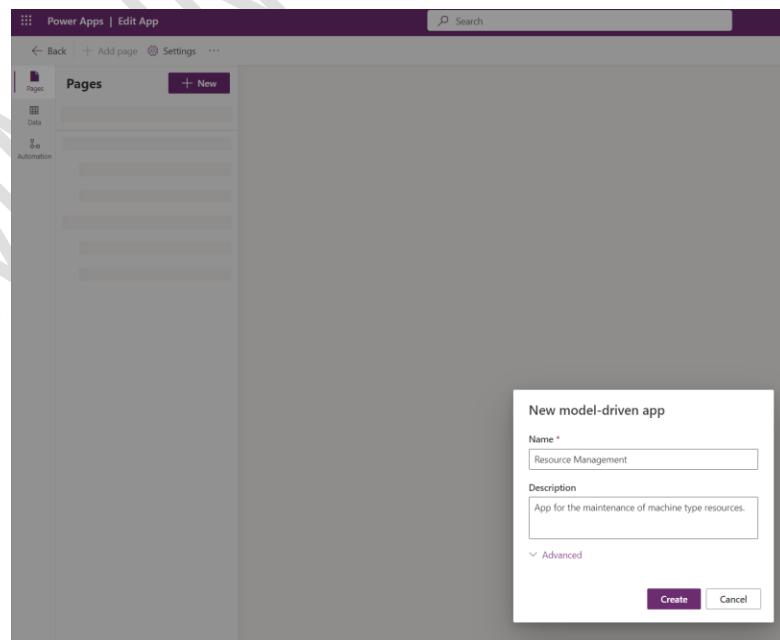


Knowing the basics, we will start creating the model-driven app:

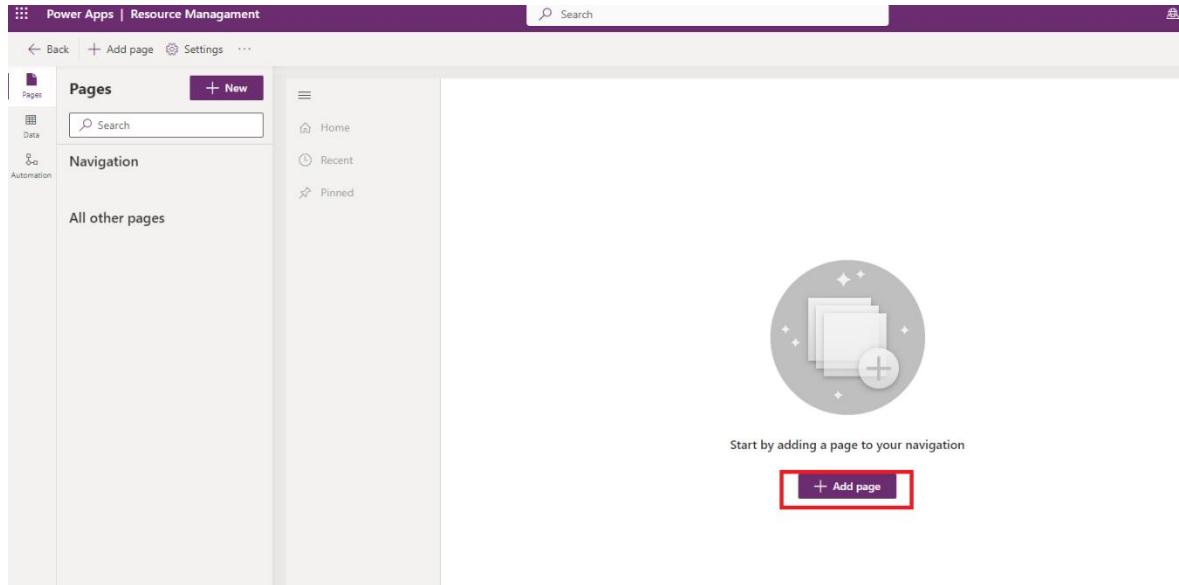
1. Open the solution, and click on New -> App -> Model-driven app.



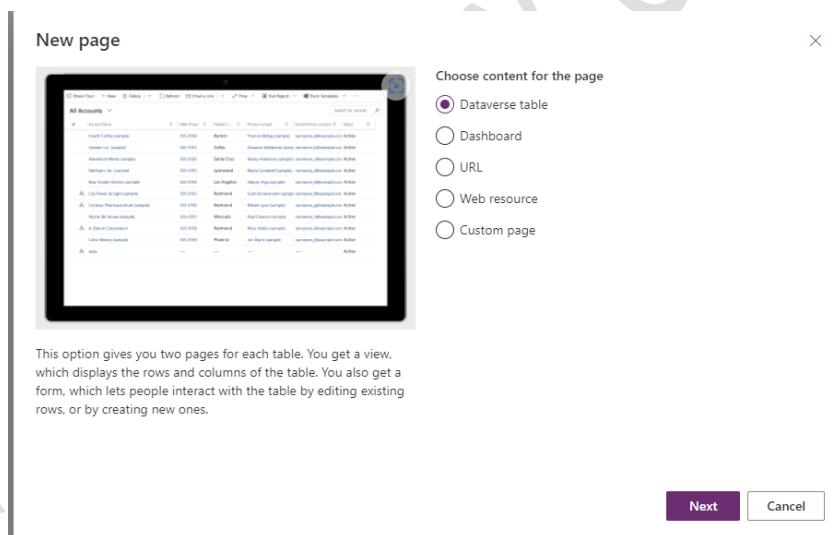
2. The model-driven app editor will open. The first thing to do is naming the app and click on create.



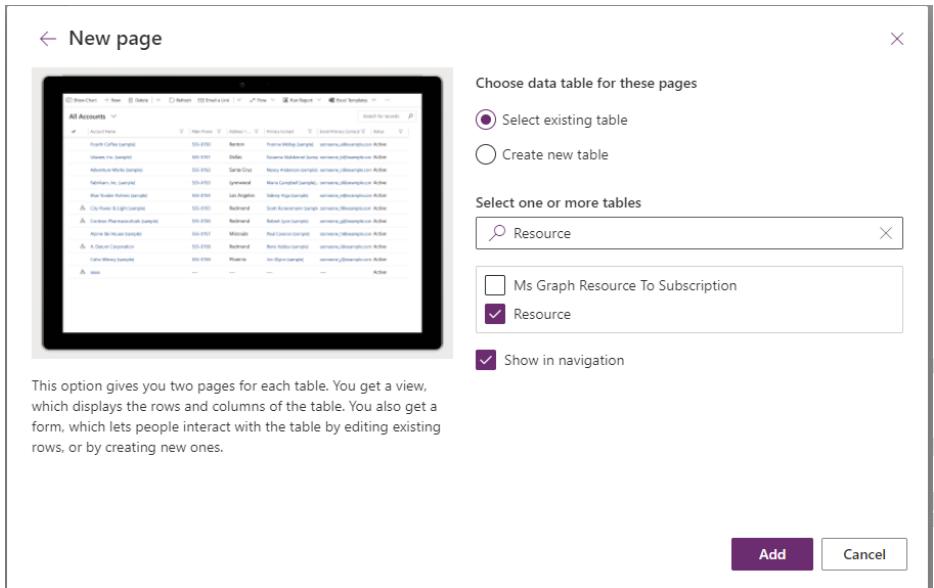
- The next step is to select the tables that we want to add to our model-driven app: click on add page.



- Select “Dataverse table” and click on Next.



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

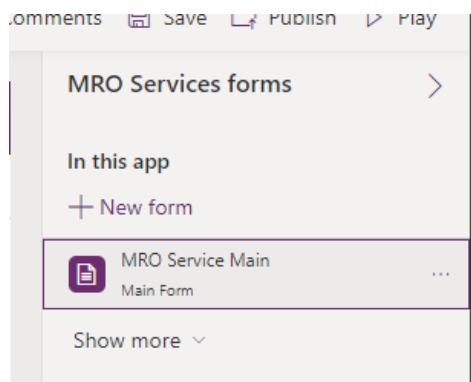


6. The tables are added as pages to our app...and we are almost ready!

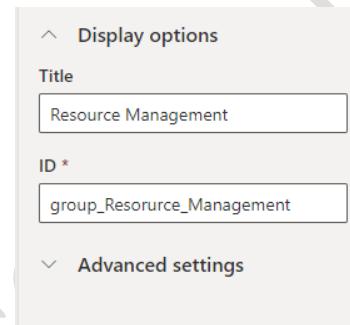
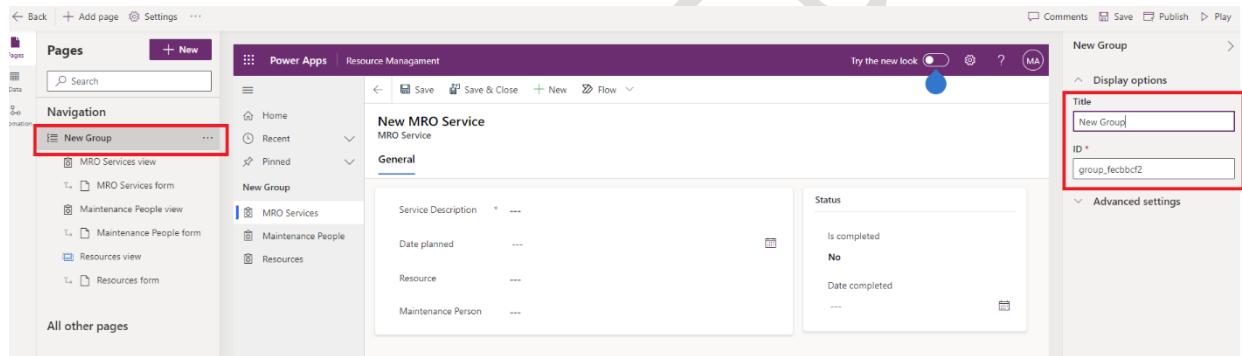
Notice that for every page shown in the navigation menu we have views and forms available.

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:



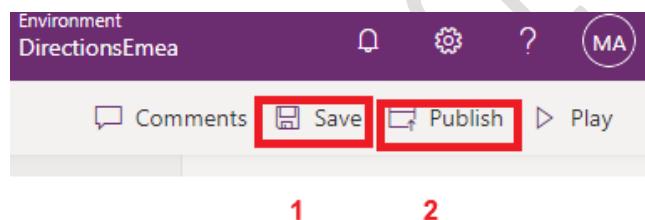
7. Click in “New Group” in the navigation pane and change the title for “Resource Management” and the ID for “group_Resource_Management.”



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 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!

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!

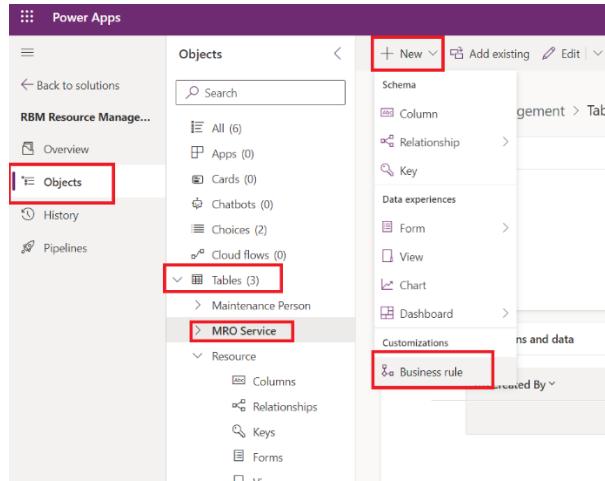
Extra Step – Developing a Business Rule (optional)

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.

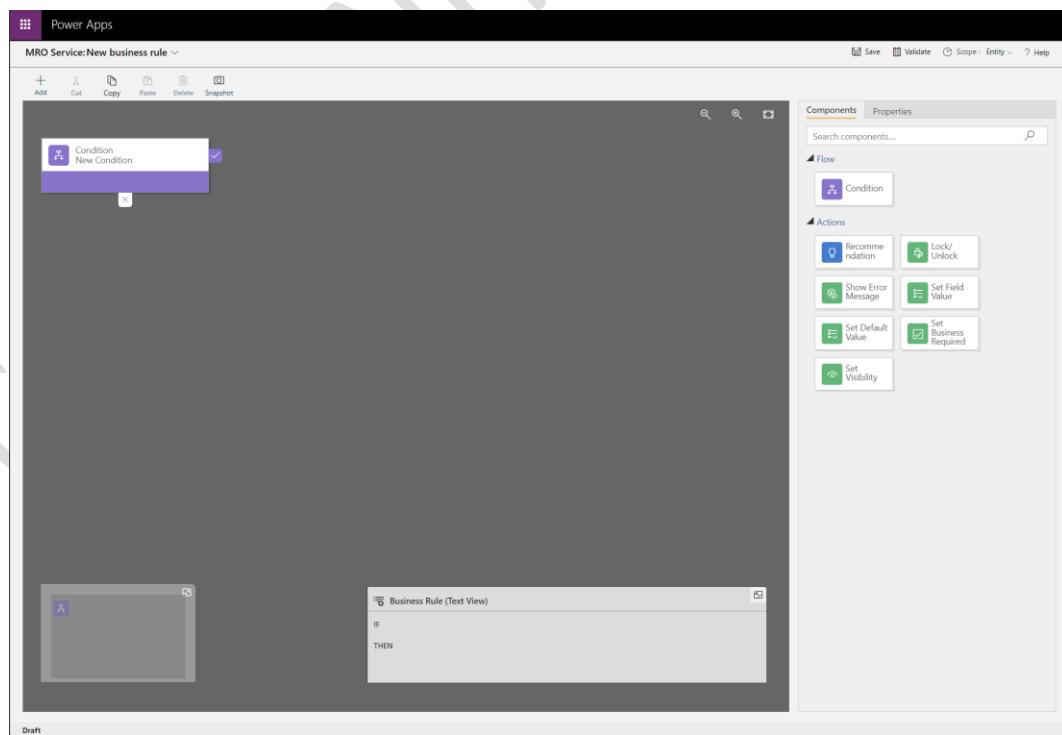
A [business rule](#) is defined in the table and allows us to implement logic without writing any code. The scope can be the table, some or all forms.

Let us start with our Business Rule:

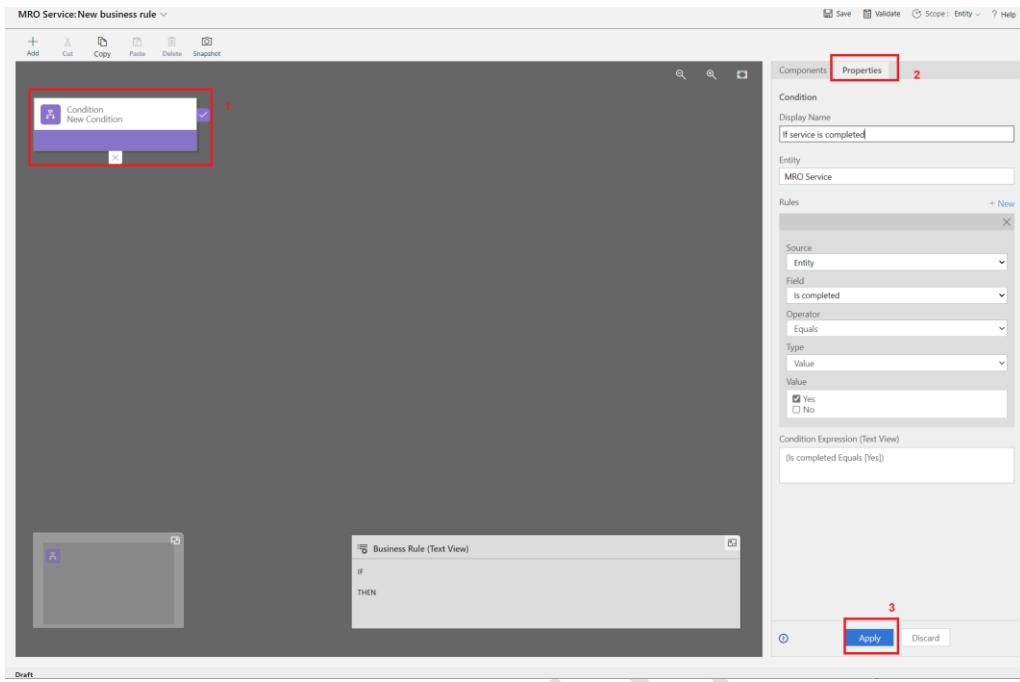
1. Select the MRO Service Table and click on New -> Customization -> Business Rule



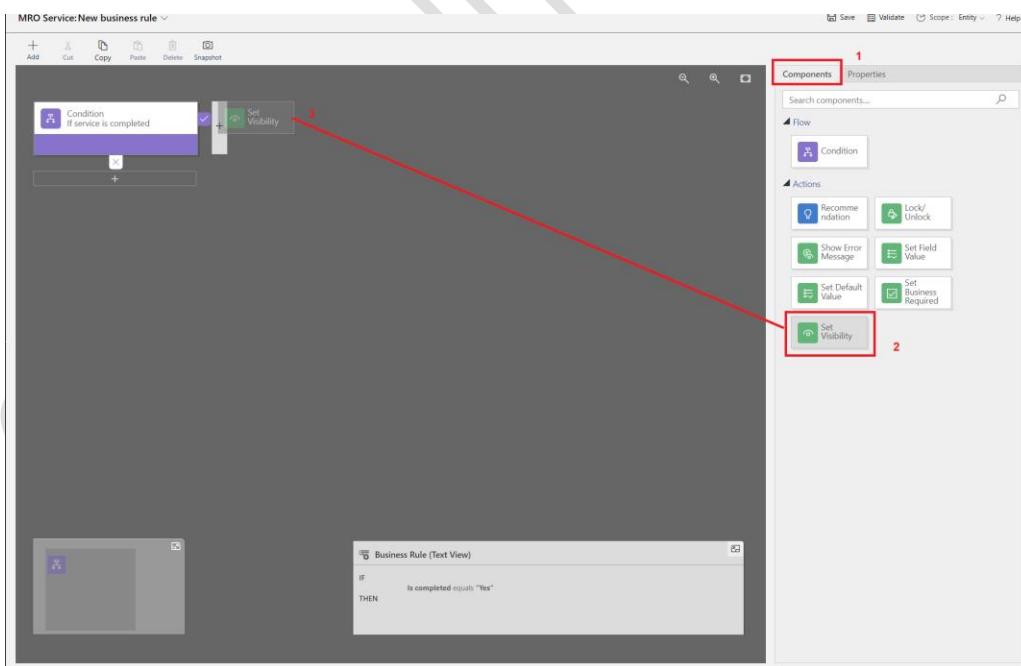
2. The business rule designer will open in a new tab:



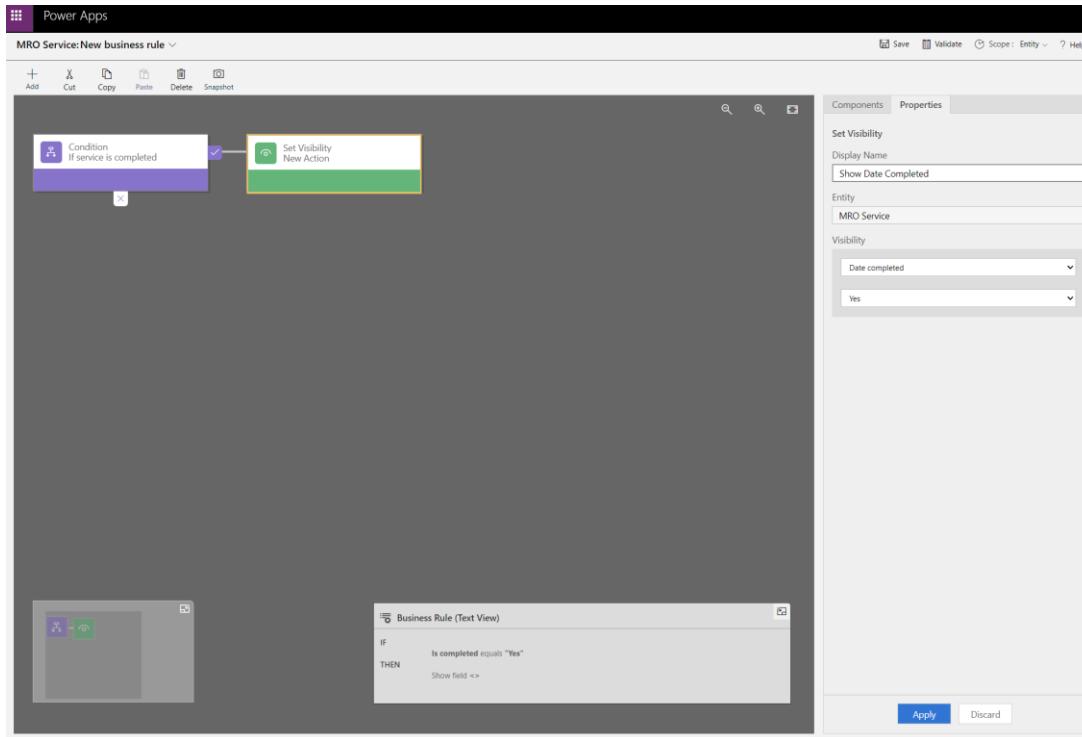
3. Click on the “Condition” box, select Properties, set the condition as show in the screenshot and click “Apply”:



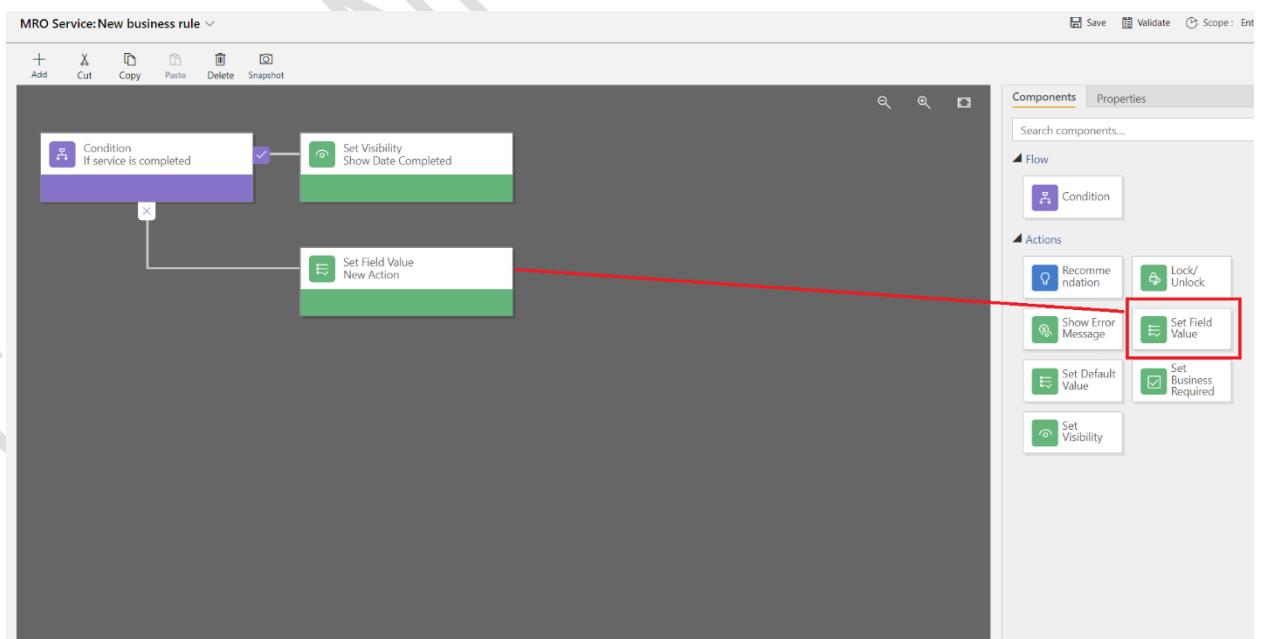
4. Click on components, select “Set visibility”, and drag and drop it to the canvas connecting it to the condition:



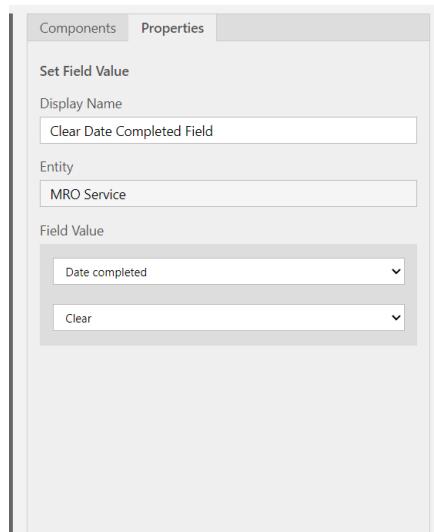
5. Click on the added box and fill the properties as shown in the screenshot, and click on apply:



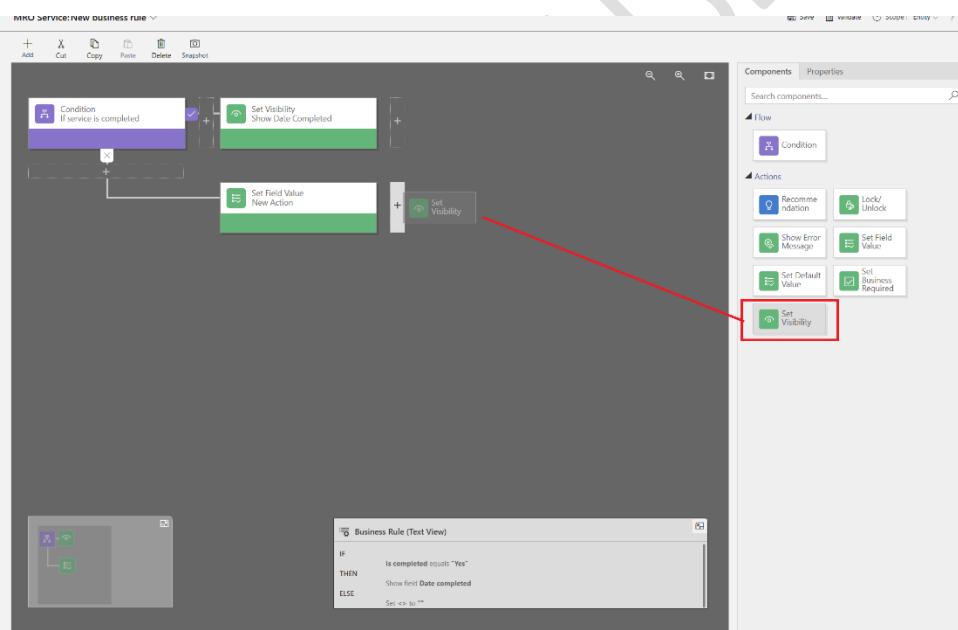
6. If the “Is completed” field, is changed to “No”, we need to clear the field and make it no visible again. Select the “Set Field Value” component.



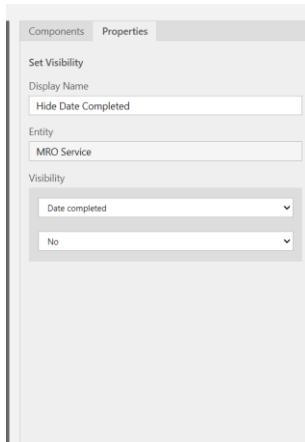
And fill the properties as shown in the next screenshot and click on “Apply.”



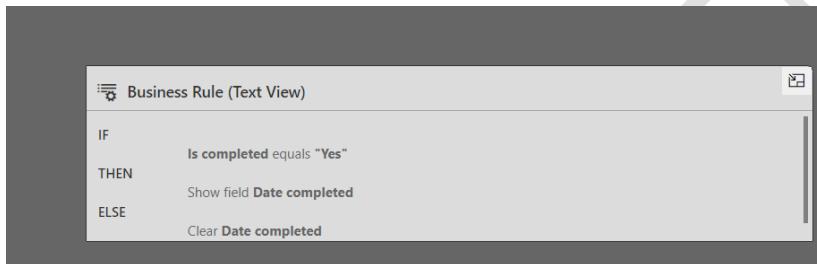
7. Select the component “Set visibility” and drop it as the next step on the “false” branch:



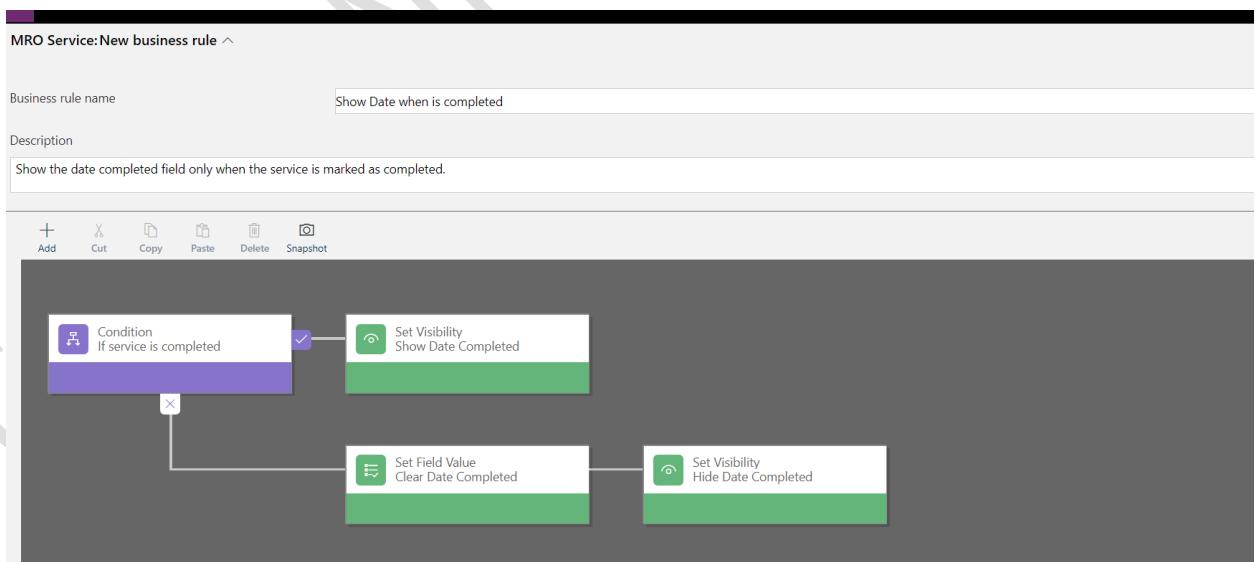
Just set up the properties as shown and click on “Apply”:



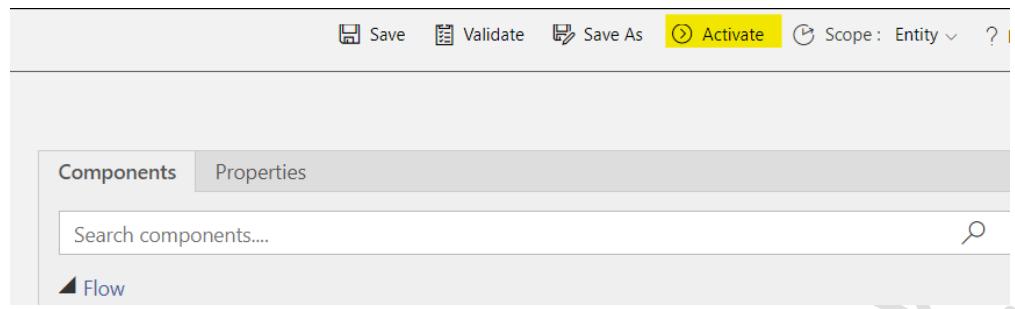
8. Now you can look at the Text View of the Business Rule and check if everything is ok.



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. The left sidebar shows objects like Apps, Cards, Chatbots, Choices, Cloud flows, and Tables. Under Tables, "MRO Service" is expanded, and "Business rules" is selected. In the main pane, a list of business rules is shown with one rule selected: "Show Date when is completed". A context menu is open over this rule, with "Turn on" highlighted. The menu also includes "Edit", "Advanced", and "Remove". The top navigation bar includes "New business rule", "Add existing business rule", "Edit", "Turn on", "Advanced", and "Remove". The breadcrumb path is RBM Resource Management > Tables > MRO Service > Business rules.

Extra Step – Try the new formula type column.

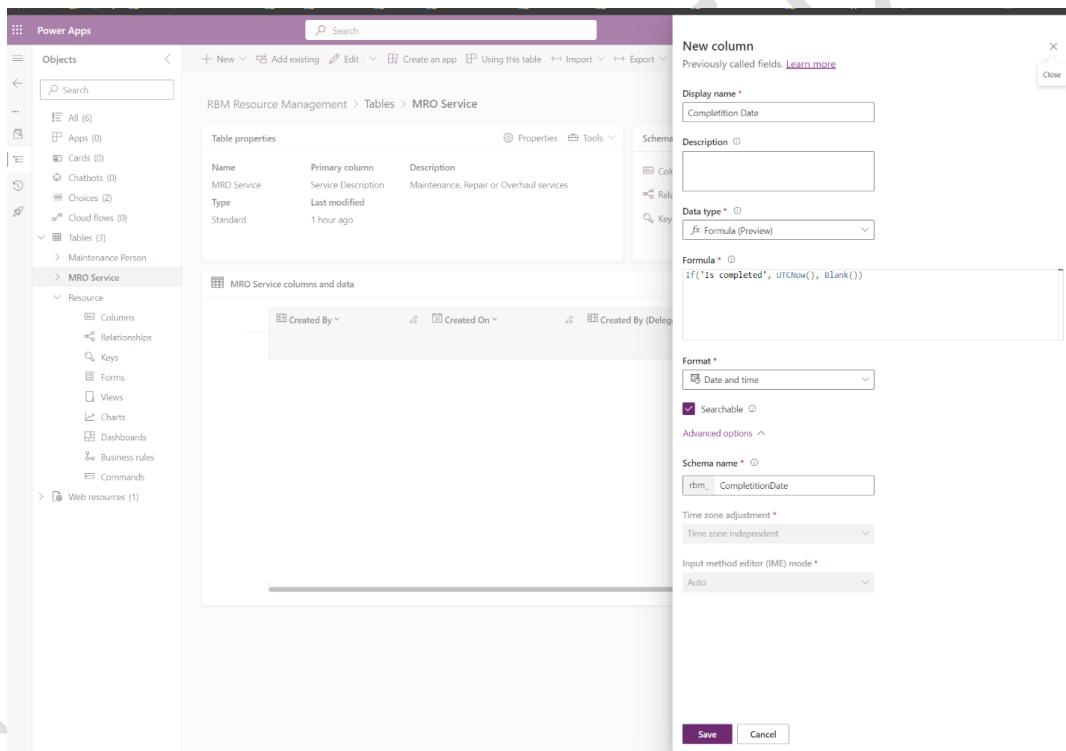
Users give us feedback and ask if it is possible to auto populate the date when the service is done when the select “Yes” in the “Is complete” field.

You can solve that, in a low code approach, using the new formula type columns. It is still a preview functionality, but you can check it! It is awesome!

<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.