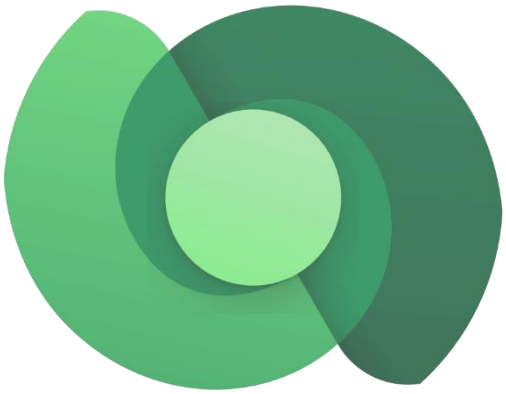
PowerTricks
Dataverse Training Labs



By Valentin Mazhar

Version History

Version	Date	Author	Comments
1.0	01/12/2023	Valentin Mazhar	First version

Table of Contents

Introduction

Welcome to the PowerTricks Dataverse Lab.

I am Valentin Mazhar and created the labs materials for this Dataverse course. I am sharing them publicly via my PowerTricks blog and GitHub repo.

I created these labs because so far Microsoft Learn only contains isolated Learning Paths about Dataverse, each addressing a different aspect with a different scenario, without delivering a single comprehensive training covering most Dataverse features. These labs are heavily inspired from some of the existing Microsoft Learn materials, such as the leading use case and data samples.

Pre-requisites

Prior experience with Power Apps and power Automate

These labs are intended for individuals who are **familiar with the Power Platform**. You do not need to be an expert but should have ideally already **created some Cloud Flows and Canvas Apps in the past**. Prior **experience with Dataverse is not required**.

If you are not familiar with Power Apps and Power Automate, you will have a better experience if you complete the free online Power Platform Fundamentals training before following these labs.

Technical set-up

To complete the labs you will need to have **access to a Power Platform Environment** with a Dataverse database provisioned. You will also need the **System Administrator role** assigned to your user in that environment.

If you do not have access to such an environment, you can contact your Power Platform Admin of your organization and share with them the "Note for Admins" document where more information is provided for them to define the appropriate way to provision this environment for you.

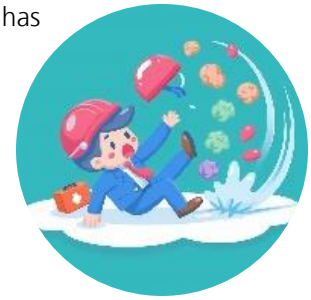
The use of Dataverse is not always allowed in organizations, so you might unfortunately be refused access. If this is the case, note that you can request your own personal M365 developer tenant for free. You will have your own tenant, independent from your organization's one and will be able to experiment there. However, these labs will not take you through the initial set up steps required to provision the tenant and your environment.

Feedback

I would love to hear your feedback about this training or anything else related to my blog. Please reach out via the PowerTricks Contact section, creating an issue in GitHub or directly via LinkedIn!

Presentation of the Use Case

Scenario



You are an employee of Contoso, an electronic manufacturing company. Contoso has been using a shared Microsoft Excel workbook to track employee accidents at its various locations. The workbook contains a table in which location managers can record data that is related to employees' accidents.

Recently, they've been having problems with data being accidentally overwritten by multiple users of the workbook, data inconsistencies in some fields, and overall performance issues with the workbook. Your manager has asked you to replace the workbook with a better solution that will eliminate these issues.

The requirements of the solution include:

- A user-friendly web interface for managers to log details about the accidents,
- Ability for managers to use dropdown menu selections to eliminate data inconsistencies,
- Ability for managers to quickly view and report on accidents,
- A notification email should be sent to the Head of Safety once the accident has been recorded and reviewed.

Based on these requirements, you have decided that a model-driven app is the best solution that will fit these requirements. You will use Microsoft Dataverse as the database to store the details about the employee accidents.

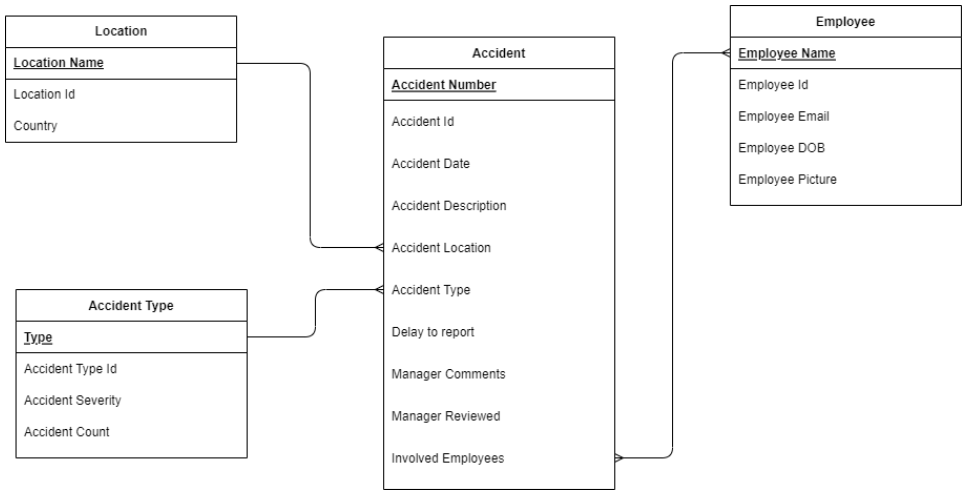
Now that you have chosen the appropriate direction for building your complete solution, you can explore the complete solution to gain a better idea of what you will be building in this training series.

Data Model

The tables below will be used to build the solution:

- **Accident**: where all the accidents will be recorded
- **Location** and **Accident Type**: these tables are used to provide configurable dropdowns when recording an accident
- **Employee**: each accident involves at least 1 employee and such involved employees should be recorded in the Accident table

The diagram below shows the different columns that will be used for each table as well as the relationships between each table.



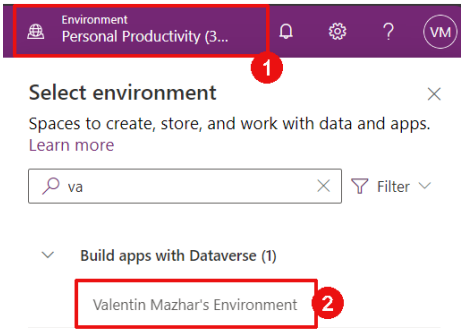
1. Explore the Complete Solution

Let's begin by importing the complete solution to explore the Dataverse capabilities. You will start creating each component of this solution one by one in the next chapter.

1.1. Explore the Environment and Dataverse

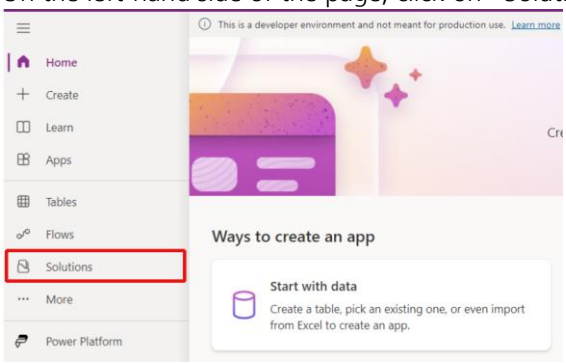
Let's first have a look at the Environment and the Dataverse tables and solutions that are provisioned by default.

1.1.1. Go to https://make.powerapps.com/home and switch to the environment that you have available for this training. For this, click on the Environment button at the top-right-hand corner of the page and select the appropriate environment:

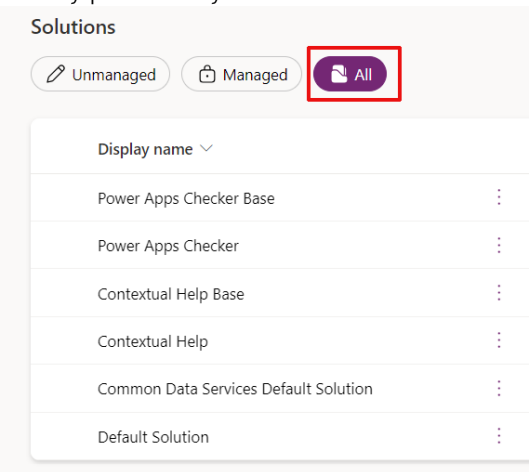


(If you are not sure which environment to use contact your Power Platform Admins and refer to the Technical set-up section of the introduction)

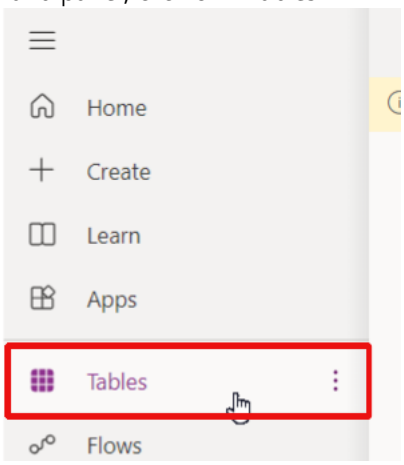
1.1.2. On the left-hand side of the page, click on "Solutions":



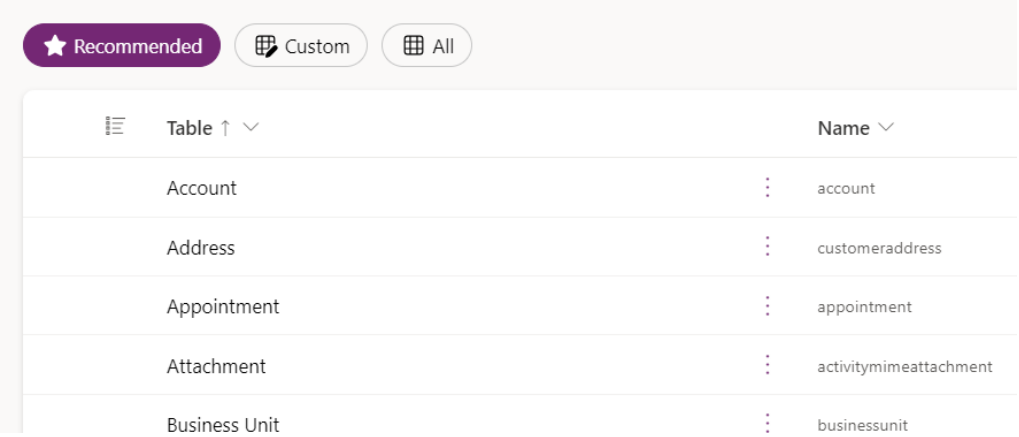
We will talk about Solutions in the next chapter, but you can see that some Solutions are already provided by default in the Environment:



1.1.3. Let's now look at the Dataverse tables already present in the environment. From the left-hand panel, click on "Tables":



(If you do not find the Tables button, you should be able to find it under "More")
You can see that several Tables are also provisioned by default on the Environment:

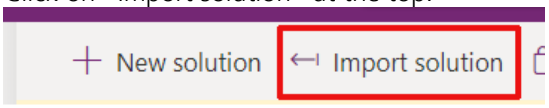


Some of these tables are used to support some functionalities. For instance, the Solutions created in the environment are stored in the "Solution" table.
Other tables are part of the Common Data Model, a set of standardized data schemas representing commonly used concepts such as the "Account", "Contact", and "Lead" tables. It is best practice to use tables when they are fit for purpose.

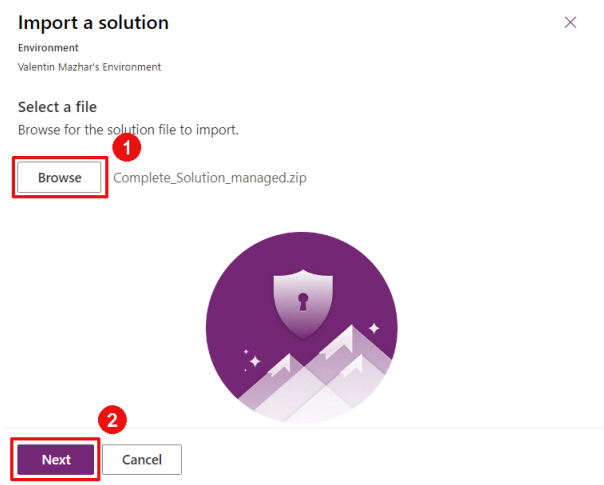
1.2. Import the complete solution

Now we are going to import the complete solution to explore the features of Dataverse and Model Driven Apps. You will create this solution step by step later in the labs.

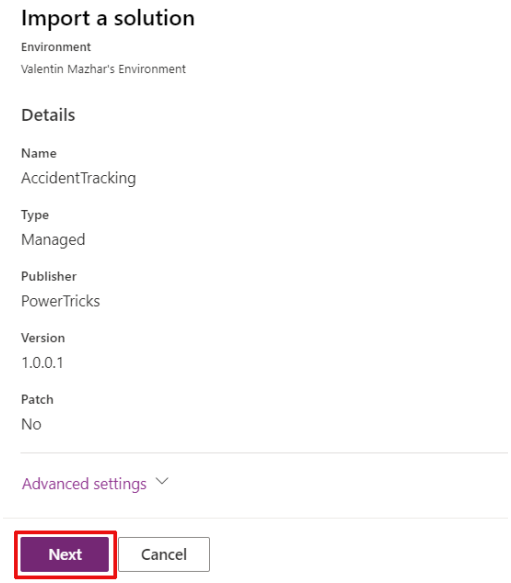
1.2.1. Open this link to download the zip folder called "Complete_Solution_managed.zip
1.2.2. Go to https://make.powerapps.com/home and switch to the environment you used in the previous chapter
1.2.3. Open the Solutions tab on the left-hand panel
1.2.4. Click on "Import solution" at the top:



1.2.5. A new panel opens on the right-hand side of the screen. Browse the zip folder downloaded in the first step. Then click "Next":

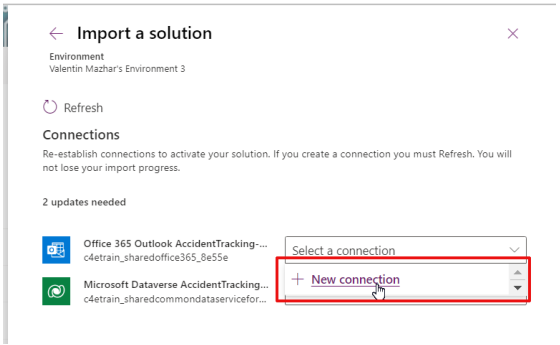


1.2.6. Once uploaded, you can find some information about the Solution you are importing (Name, Publisher, etc…). We will talk more about this in the next chapter. Click Next:

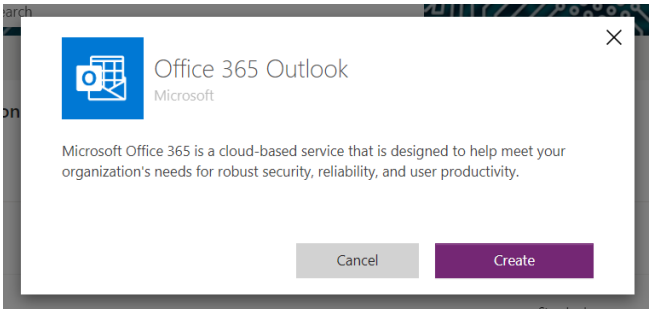


1.2.7. If you have never used the outlook and dataverse connectors on this environment before, you will need to create the connections. This is because the Solution contains a Flow that requires these connections. For each connection:

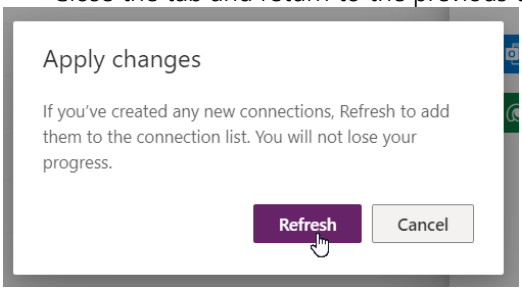
Click "New connection"



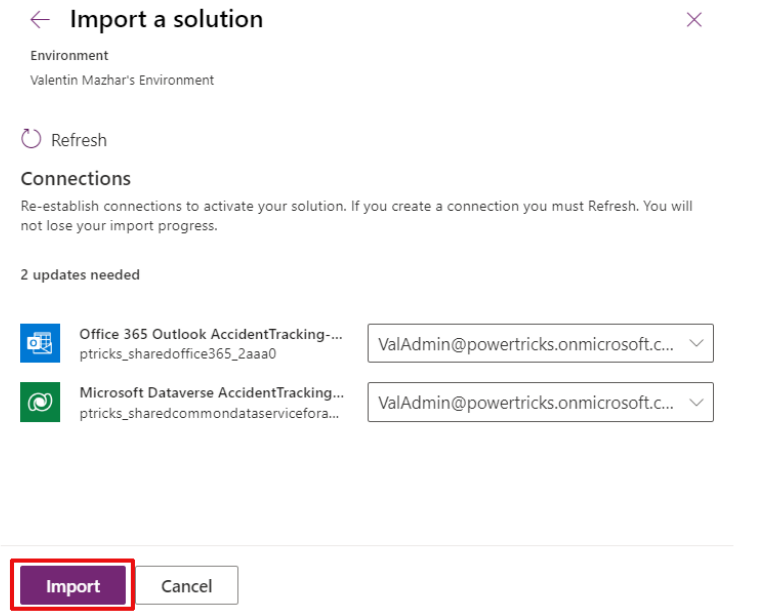
A new tab is open, click "Create" and authenticate with your account to create the connection:



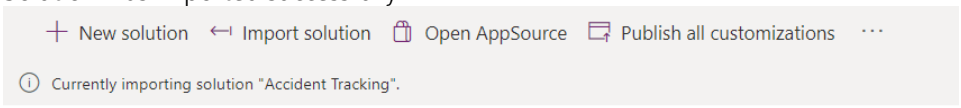
Close the tab and return to the previous tab and click "Refresh":



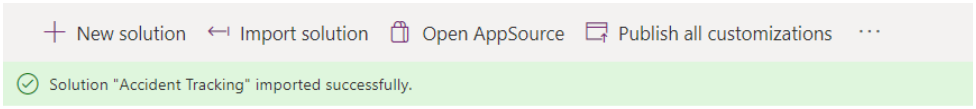
1.2.8. Once this is done for both connections, click "Import":



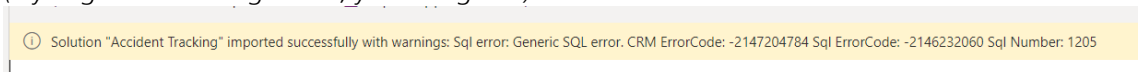
1.2.9. You should see a horizontal grey banner indicating "Currently importing solution 'Accident Tracking'. It can take a few minutes. Wait until you see a green message indicating that the Solution was imported successfully:



…

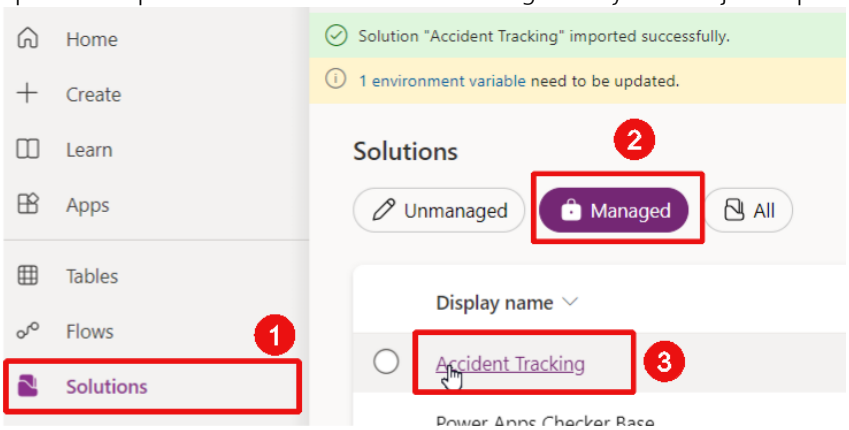


(If you get the warning below, you can ignore)

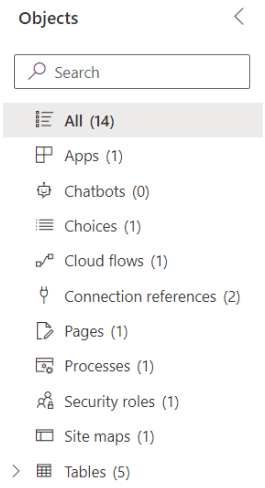


1.3. Explore the Accident Tracking Solution

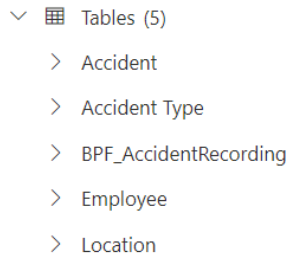
1.3.1. Open the imported solution "Accident Tracking" that you have just imported:



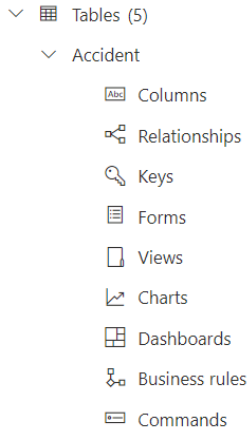
1.3.2. The solution is now and you can see on the left a tree view of all the components included in that Solution:



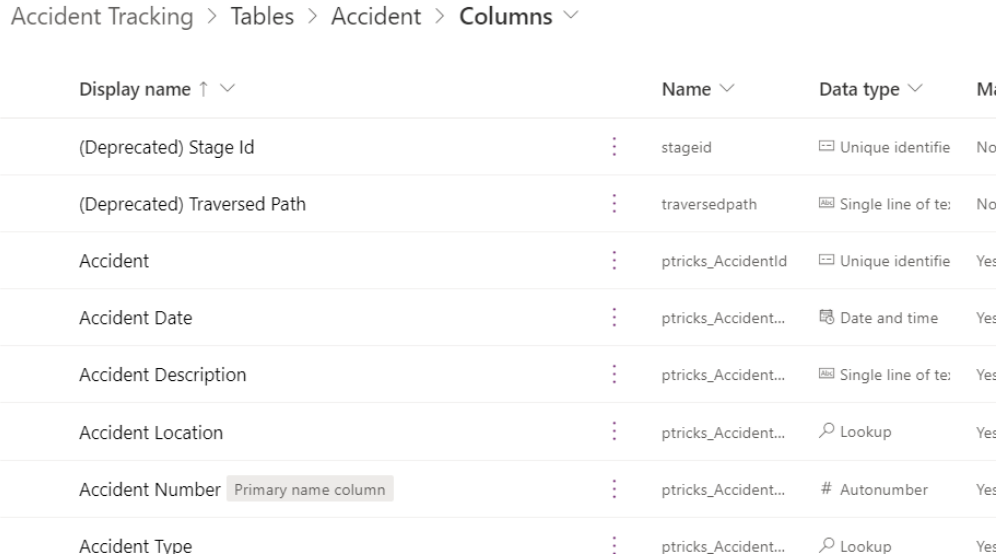
1.3.3. Extend the "Tables (5)" item, you will see all the Tables included in the Solution. These tables are described in the Data Model section at the beginning of this document. Except one table that will be discussed in another chapter:



1.3.4. Extend the "Accident" table in the tree view. You can see what type of metadata is associated to this table, such as columns, relationships, charts, forms and views.



1.3.5. From this extended menu, click on "Columns". All the columns of the Accident table are then listed on the screen. These columns are composed of default columns created by default (such as Created By) and custom columns created for this solution. All the custom columns have their "Name" starting with "ptricks_":

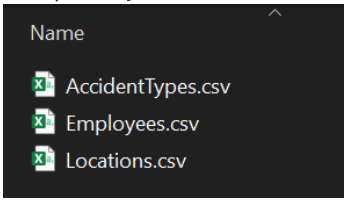


1.4. Import data into the tables

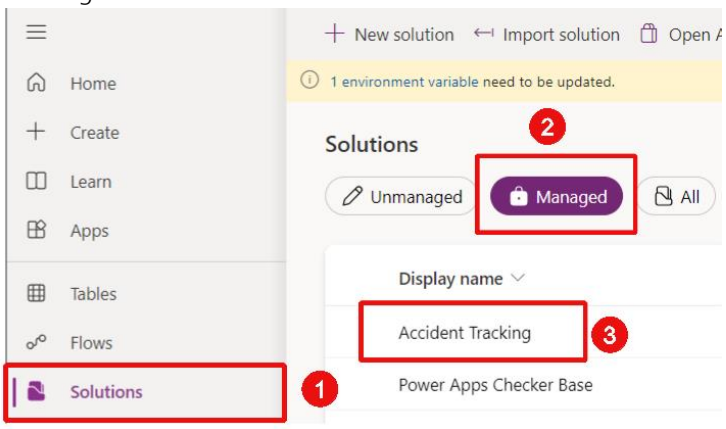
Before exploring the App included in the solution, let's first import some data into the Location, Accident Type and Employee tables. This will allow us to save some time by doing some bulk import as opposed to manually inputting the data row by row.

1.4.1. Open this link and download the zip folder "Dataverse Data Import.zip"

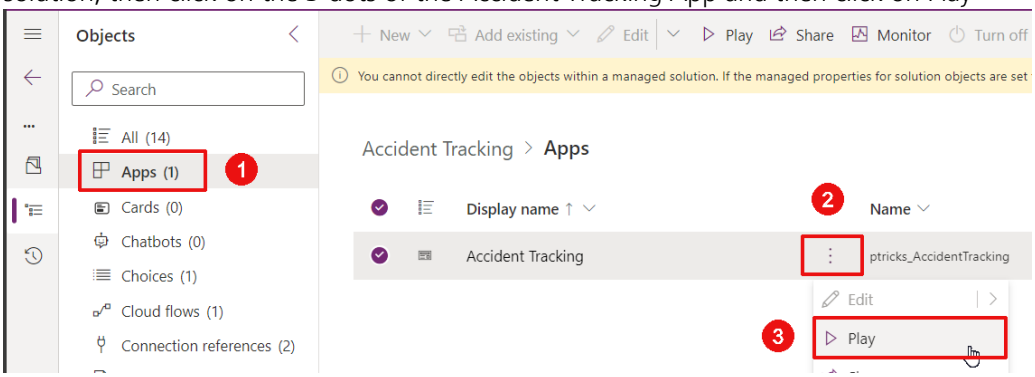
1.4.2. Unzip it in your download folder, inside you will find 3 csv files:



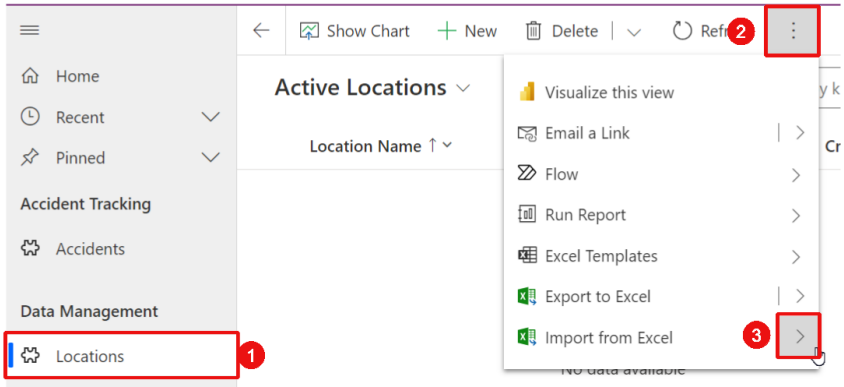
1.4.3. Open https://make.powerapps.com/ and make sure that you are in the environment used in the chapter 1.1. Open the Solutions tab on the left panel and click on the "Accident Tracking" solution:

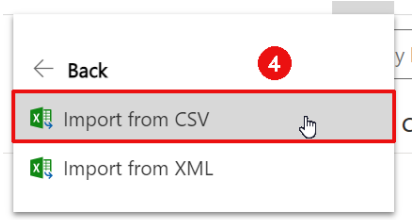


1.4.4. On the left-hand panel, select "Apps (1)" to filter and show the App included in the solution, then click on the 3 dots of the Accident Tracking App and then click on Play

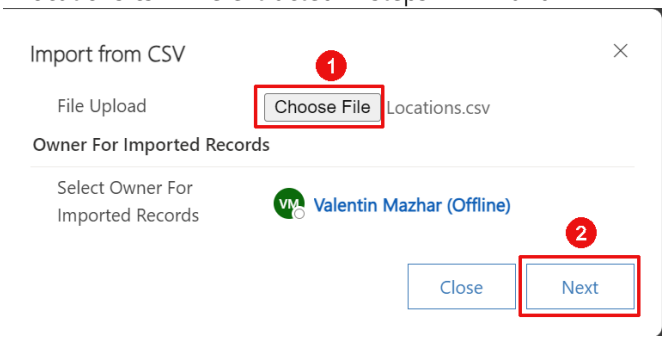


1.4.5. Navigate to the "Locations" page on the left. Then from the top command bar, click on "Import from Excel" > "import from CSV". If you do not see the button you might need to click on the 3 dots first:

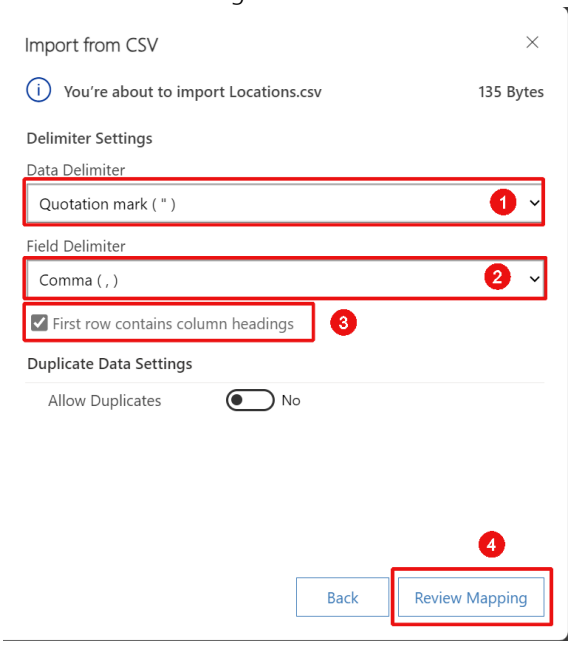




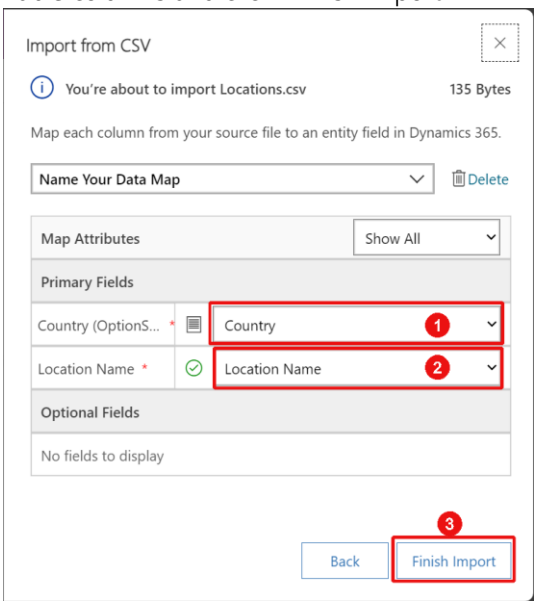
1.4.6. From the right-hand panel which has appeared, click on "Choose File" and select the "Locations.csv" file extracted in steps 1.4.1. and 1.4.2. Then click "Next":



1.4.7. Make sure the configuration is as shown below and click "Review Mapping":

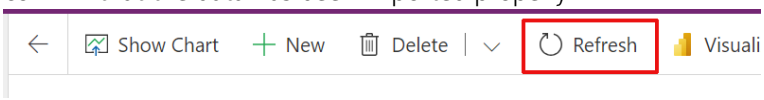


1.4.8. Confirm that the Columns from Excel have been automatically mapped correctly with the Table columns and click "Finish Import":

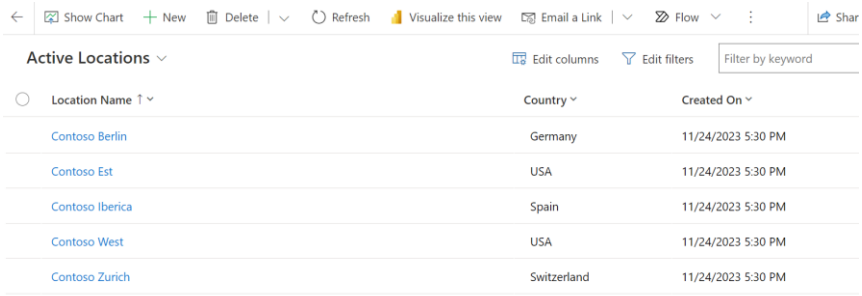


1.4.9. Click "Done":

1.4.10. You might need to wait a few minutes, then click "Refresh" from the top command bar to confirm that the data has been imported properly:



1.4.11. Note that 5 rows have been imported successfully:

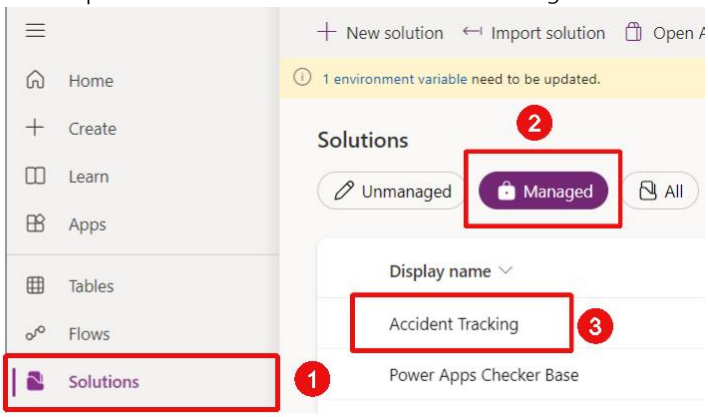


1.4.12. Reproduce the same steps for the Employees table
1.4.13. Reproduce the same steps for the Accident Types table

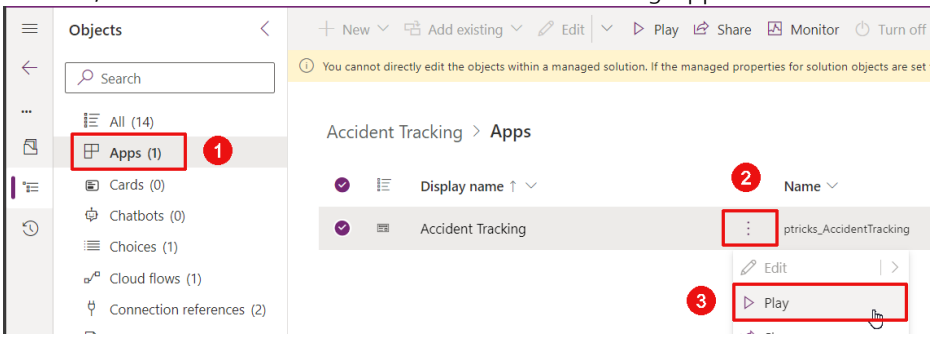
1.5. Explore the Model Driven App

Now that we have some data imported, let's explore the Application further.

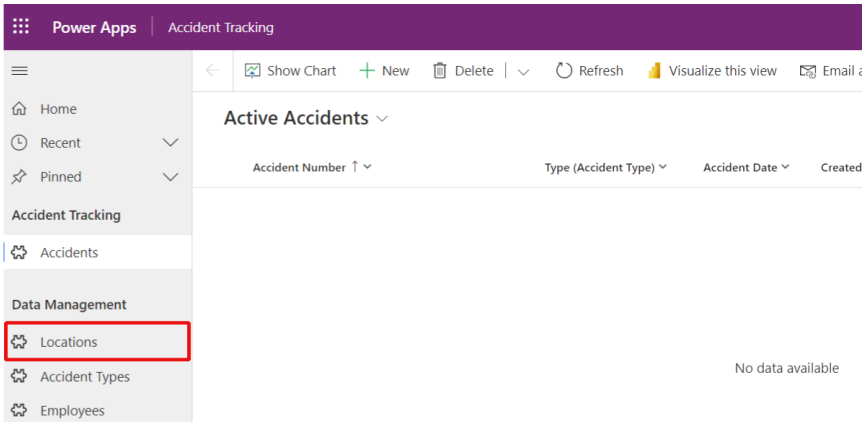
1.5.1. If you no longer have the window open, open https://make.powerapps.com/ and make sure that you are in the environment used in the chapter 1.1. Open the Solutions tab on the left panel and click on the "Accident Tracking" solution:



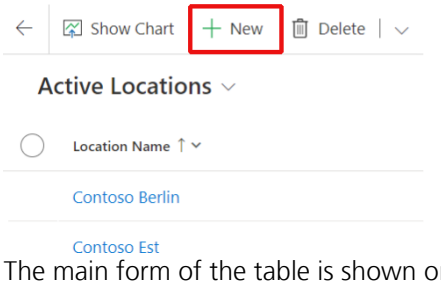
On the left-hand panel, select "Apps (1)" to filter and show the App included in the solution, then click on the 3 dots of the Accident Tracking App and then click on Play:



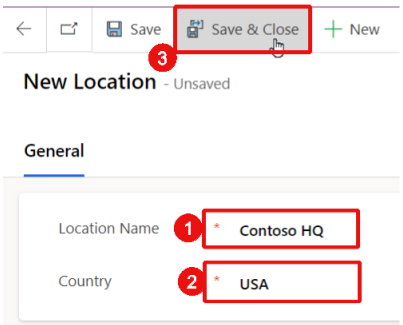
1.5.2. The App opens on the Accidents page and we can see that there is no data available. This is normal as we did not import any accidents. Open the "Locations" view by clicking on "Locations" on the left:



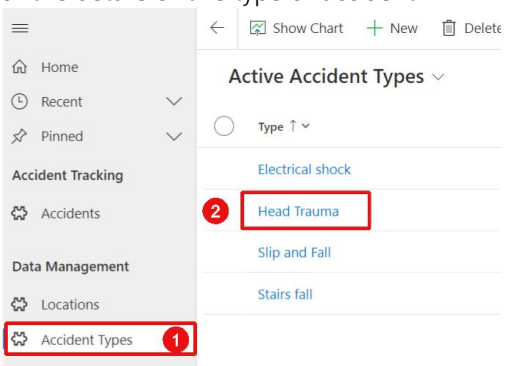
1.5.3. Some locations are then listed on the page view. They are the ones that you imported from the file earlier. From the command bar at the top, click "New" to add a new Location:



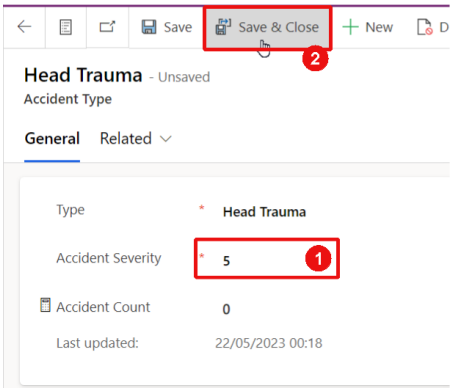
1.5.4. The main form of the table is shown on the screen. Add "Contoso HQ" for the Location Name and select USA for the Country. Then click "Save & Close" from the command bar:



1.5.5. Open the "Accident Type" view. As for the Locations, you can find 4 rows already populated from the import done earlier. Click on "Head Trauma" to open the main form of the details of this type of accident:



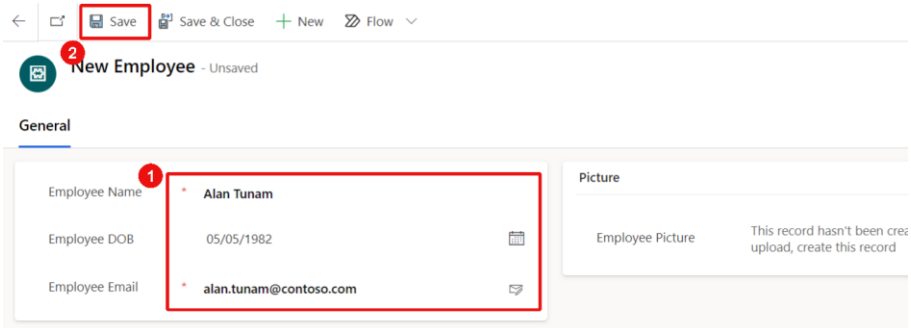
1.5.6. Change the severity from 4 to 5 and click "Save & Close":



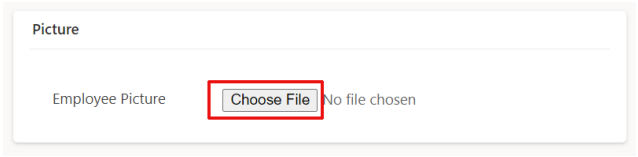
1.5.7. Open the Employees view by clicking on "Employees" on the left-hand panel. And click "New" and add the Employee the details below:

Employee Name: Alan Tunam
Employee DOB: 05/05/1982
Email: Alan.Tunam@contoso.com

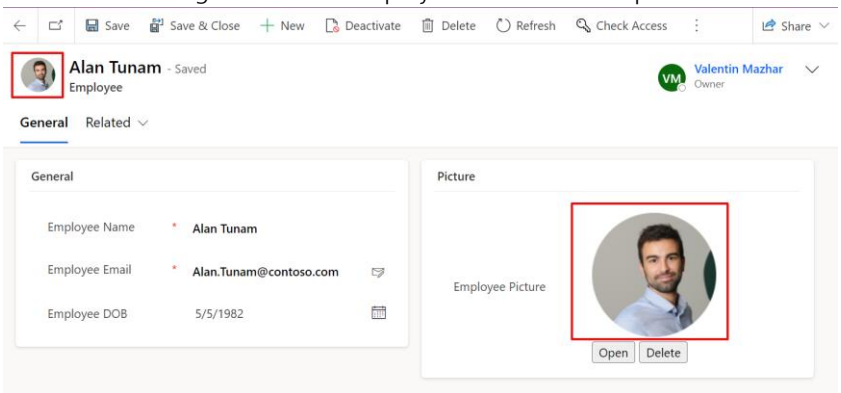
Click "Save":



1.5.8. Once the record has been created it is possible to add an Employee Picture. Click "Choose File" on the Picture section:

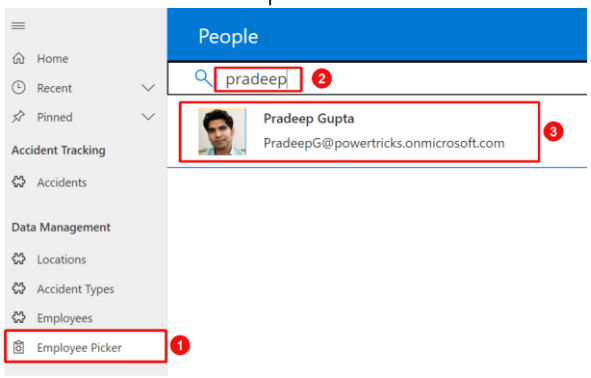


1.5.9. Select an image from your computer and click Save again. You can see that the image is now also showing next to the Employee name at the top:



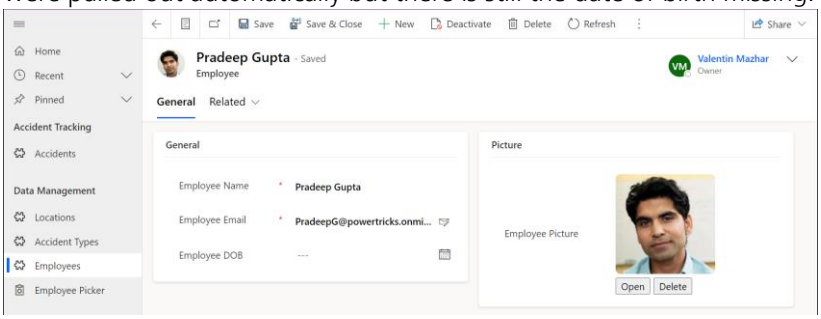
1.5.10. Now open the "Employee Picker" page from the left-hand Panel and click on "Allow" to allow the Office 365 Users connection. This is a custom page that was created as a Power

Apps canvas App to make the process to add a new employee from the company easier and reduce human error during manual input. Search for your account or a colleague's in the search bar at the top and select it:



1.5.11. Once selected, click the submit icon on the top-right hand corner:

1.5.12. Go back to the Employees view. You can now see a new Employee added with the appropriate name. If you open the record you will see that the email address and picture were pulled out automatically but there is still the date of birth missing:



1.5.13. Let's record a few accidents. Open the Accidents view (left-hand panel) and click "New". Add the details below:

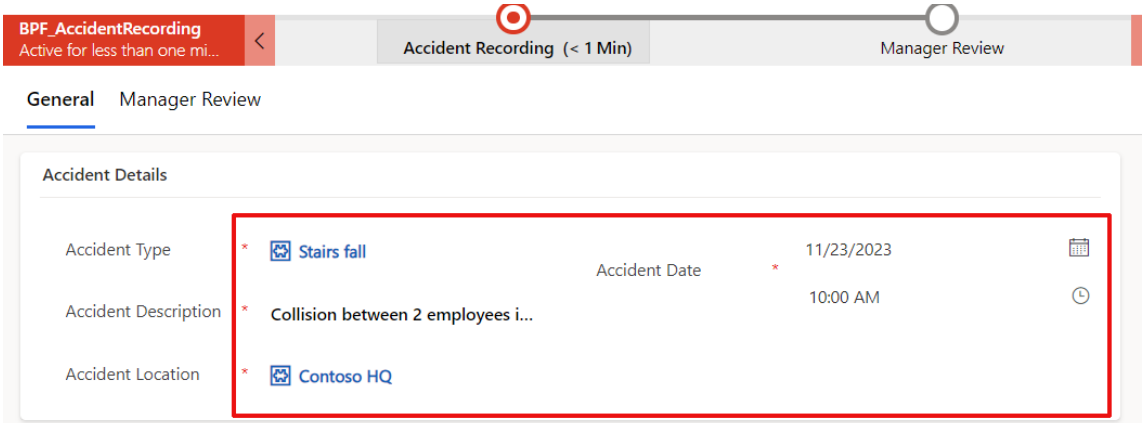
Accident Date: November 23rd 2023 at 10:00

Accident Description: Collision between 2 employees in the stairs. One of them broke their wrist.

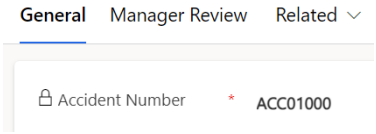
Accident Type: Stairs fall (select the field and press Enter to see all the options available)

Accident Location: Contoso HQ (select the field and press Enter to see all the options)

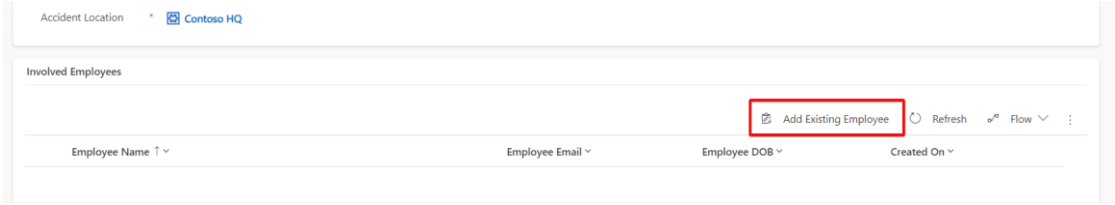
Click save:



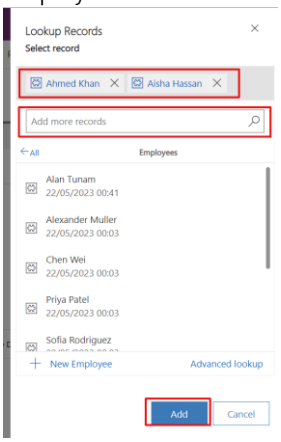
1.5.14. Note that saving the accident automatically created and show the Accident Number:



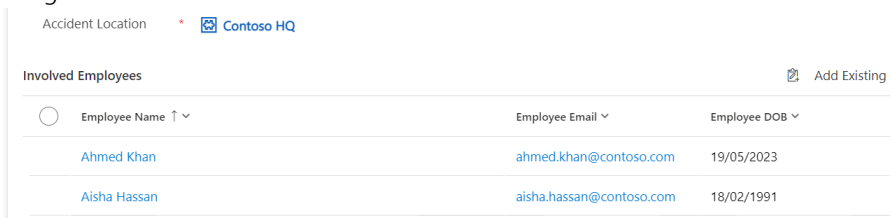
1.5.15. A new subgrid called "Involved Employees" also appears at the bottom of the screen. This will allow us to add the employees involved in the accident. Click on "Add Existing Employee":



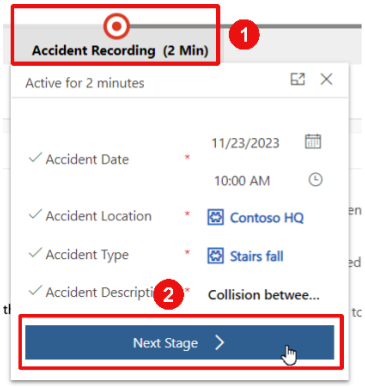
1.5.16. A panel opens on the right. Click on "Look for records" and press Enter to see the list of employees. Select "Ahmed Khan" and "Aisha Hassan" and click "Add":



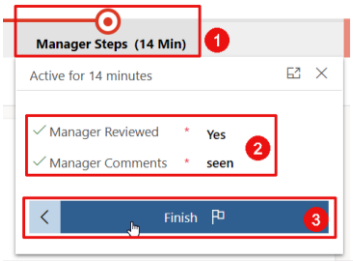
1.5.17. Click "Refresh" next to "Add Existing Employee" to see the employees listed in the subgrid:



1.5.18. On the progress bar at the top, click on "Accident Recording" and then "Next Stage" to confirm that this stage of the process has been completed:

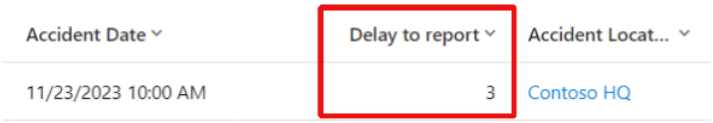


1.5.19. You can then finalize the process by clicking on "Manager Steps" on the progress bar, updating "Manager Reviewed" to "Yes", adding "seen" for "Manager Comments" and click "Finish":

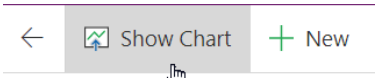


1.5.20. You should receive a confirmation email, check your inbox

1.5.21. Go back to the Accidents view by clicking on the "Accidents" tab on the left-hand panel. You can see the data related to the incident. Note the column "Delay to report" that is automatically calculated and show the difference in full days between the Accident Date and the Creation Date:

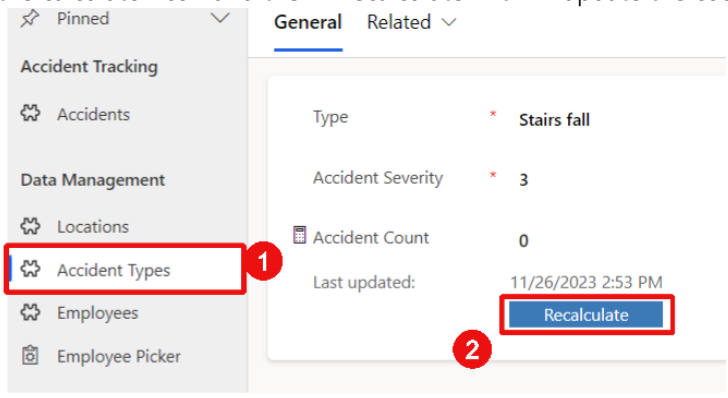


1.5.22. From the Accidents view, click on "Show Chart" at the top:



1.5.23. A chart is shown on the left of the view to show the count of accident per location. It is also possible to change to another chart to show the accident per accident type:

1.5.24. Open the Accident Types view on the left panel and open the "Stairs fall" details. Click on the calculator icon and then "Recalculate". It will update the count value to 1:



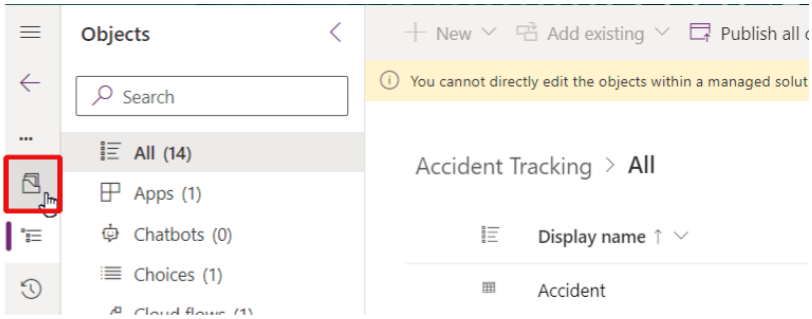
This is a "Roll Up" column that calculates the count of accidents associated to this accident type. It is regularly automatically updated but can be manually updated to get the most accurate information.

This section provided an overview of the main functionalities that will be implemented in the following chapters. Feel free to continue experimenting with the application to get familiar with the interface.

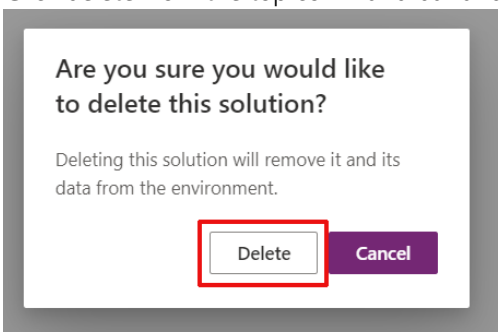
1.6. Delete the Solution

Let's now delete this Solution since we want to create it from scratch during the following sessions.

1.6.1. From the "Accident Tracking" Solution page, click on the overview icon on the very left of the screen:



1.6.2. Click delete from the top command bar and confirm "Delete":



Since it is a "Managed" solution, deleting it will automatically delete all the underlying components which will clean up your environment for the next chapter.

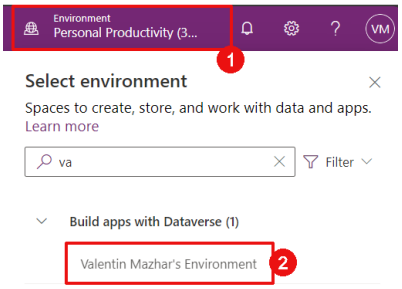
2. Create the Data Structure

Let's start by creating the data structure of the solution tested in the previous chapter.

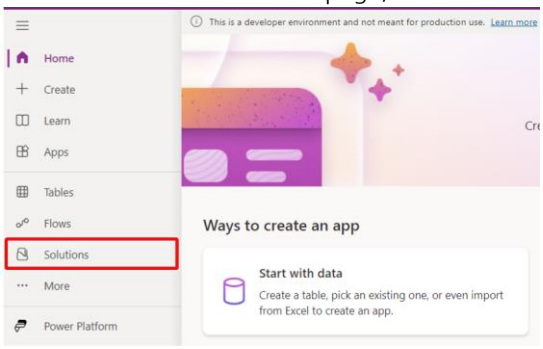
2.1. Create the solution

A "Solution" acts as a package which can contain Power Platform objects such as Dataverse Tables, Power Apps Apps, Power Automate Flows, etc. It is a best practice to start by creating a Solution and then create the necessary objects inside that solution.

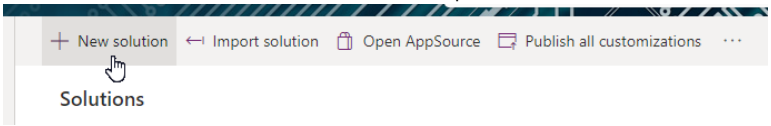
2.1.1. Go to https://make.powerapps.com/home and switch to the environment that you have available for this training. For this, click on the Environment button at the top-right-hand corner of the page and select the appropriate environment:



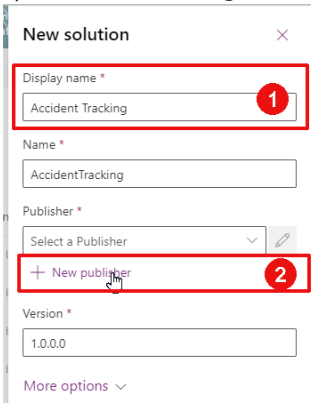
2.1.2. On the left-hand side of the page, click on "Solutions":



2.1.3. Click "+ New solution" from the top horizontal banner:

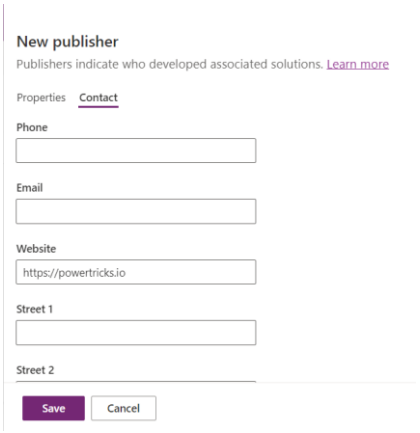


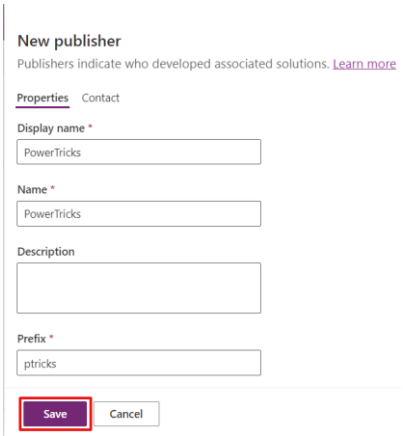
2.1.4. On the right-hand panel that is automatically open on the screen, add "Accident Tracking" for the Display Name. Notice that the Name is automatically created without any space. Do not change it. Then, click on "+ New publisher":



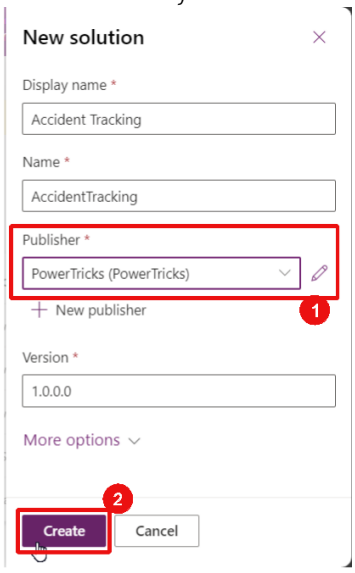
2.1.5. Let's now create a publisher for the solution. It should identify who owns that solution. For the sake of this exercise, we will be creating a publisher as below:

Display name: PowerTricks
Name: PowerTricks
Prefix: ptricks
Website (Contact tab): https://powertricks.io
Once done, click "Save" at the bottom of the panel





2.1.6. Select the newly created Publisher and click "Create":



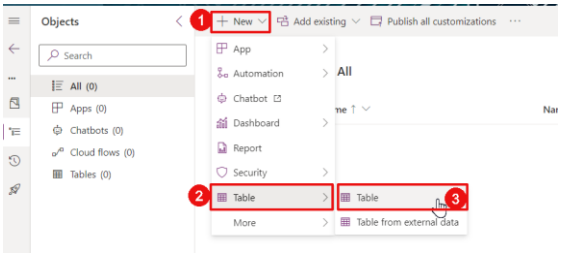
2.2. Create the Location table

Now that the solution is created with an appropriate publisher, let's start by creating our first table: Location. Each work accident will be associated to a company location stored in this table.

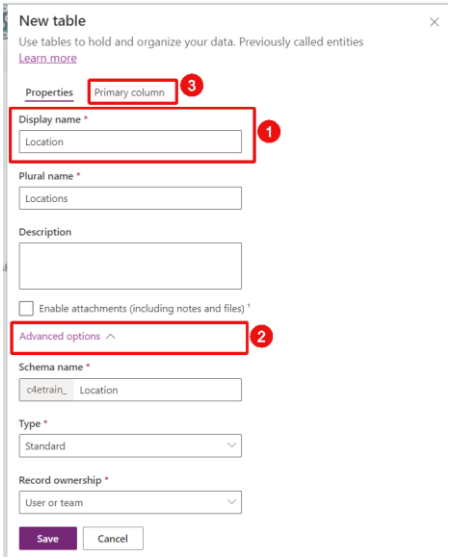
2.2.1. Go to https://make.powerapps.com/home and switch to the environment that you have available for this training.

2.2.2. On the left-hand panel, click on "Solutions"

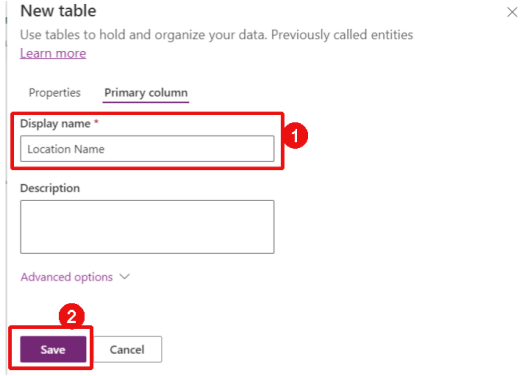
2.2.3. Open the "Accident Tracking" solution created earlier. Then click "+ New" > "Table" > "Table":



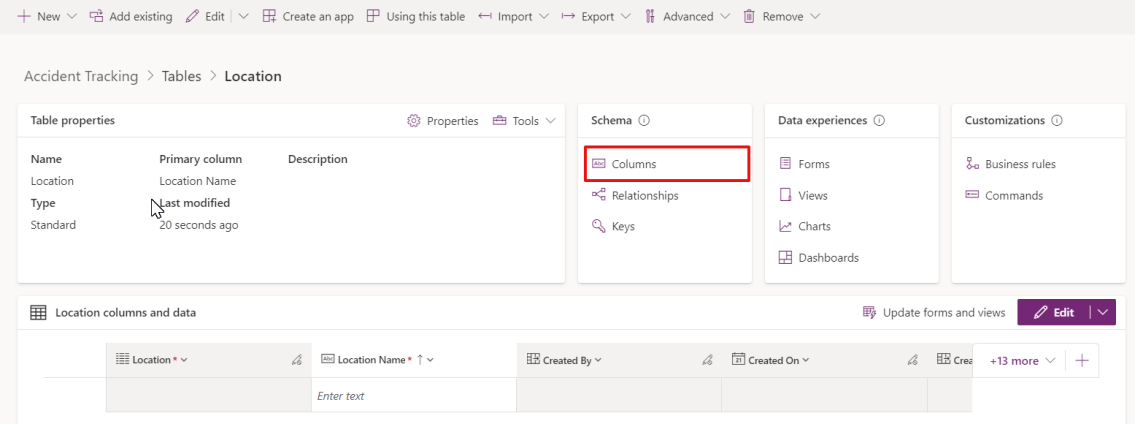
2.2.4. On the right-hand panel, add "Location" for the Display Name. Notice that the Plurial name is automatically added, do not modify it. You can also explore the Advanced options (without modifying them). Then click "Primary column":



2.2.5. The Primary column will define how you identify a row from the table. This is the information that will be shown when selecting it as a lookup field from another table, and therefore should be meantingful and representative of the data stored in the table. Add "Location Name" for the Display name of the primary column and click "Save":



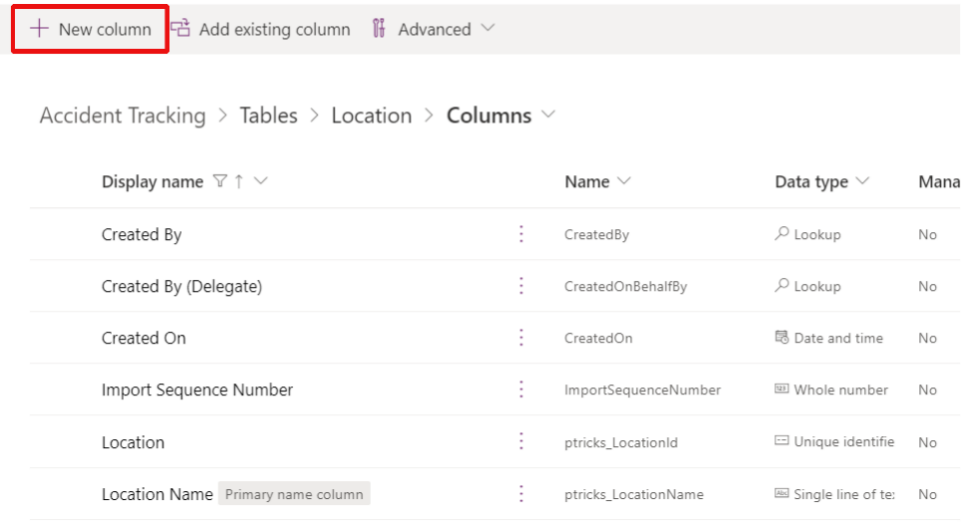
2.2.6. The table is now created and a Table overview page opens on the sreen. We now need to create the required columns. Click on the Columns button from the Schema section:



2.2.7. Notice all the columns created by default (you might need to refresh the page to see them). These columns include:

- Location Name (technical name "ptricks_LocationName"): the Primary column defined earlier
- Location (technical name "ptricks_LocationId"): the unique identifier (GUID) automatically created for each row
- Other columns such as Created By, Created On, etc.

Click "New column":



2.2.8. We would like to add a Country column for users to select from a dropdown. This will correspond to the country of the office where each accident occurred. There are 2 main options to propose a dropdown to users: with a Choice column or a Lookup column.

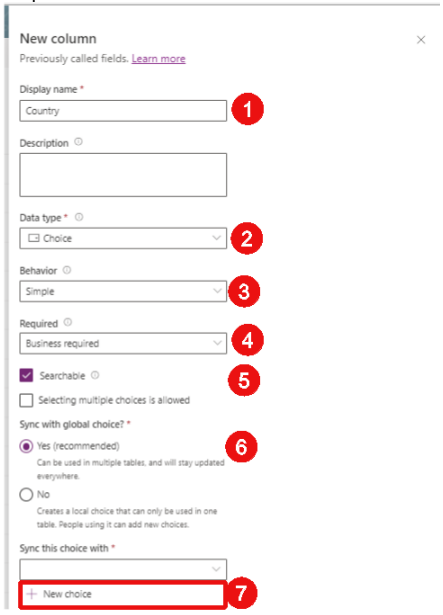
- Choice column: This is defined from Dataverse, usually by the solution Maker. It is suitable when the information does not need to change often, and if a single value is sufficient for the purpose of the column.
- Lookup column: the values proposed for that column are sourced from another table. This is suitable when the information can change often and needs to be maintained by the users of the application. It is also preferable when a lot of information should be associated.

In our case, the countries will rarely change and we do not need to have more information than the name of the country, therefore we will create a Choice column.

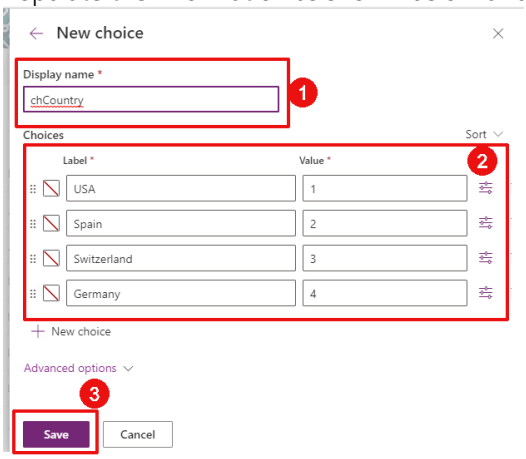
There are two main options to create a Choice column: create a local choice or a global choice:

- Local choice: a local choice is only available in the current table
- Global choice: a global choice can be used in multiple tables.

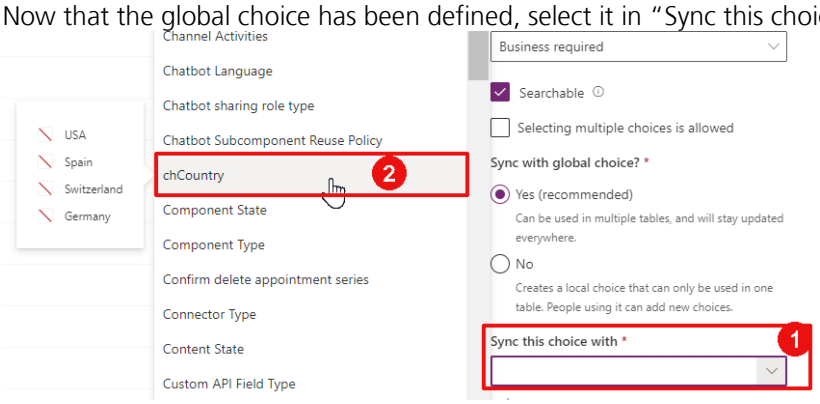
In our case, it is likely that the office countries will be required by other solutions in the future, so we will create a global choice.
Populate the New column form as shown below, and click "+ New choice":



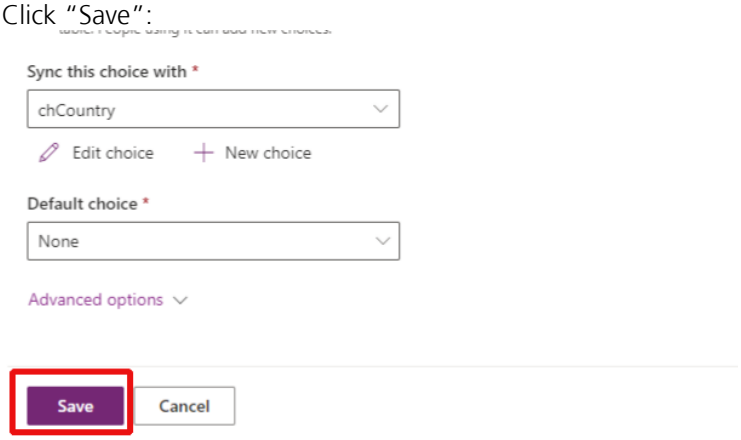
2.2.9. Populate the information as shown below and click "Save":



2.2.10. Now that the global choice has been defined, select it in "Sync this choice with":

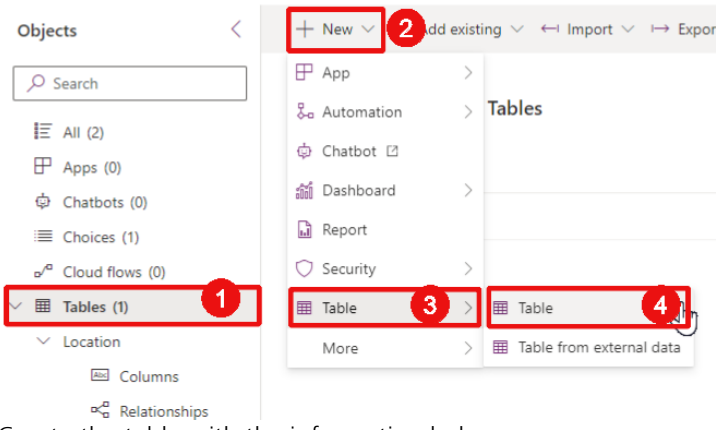


2.2.11. Click "Save":



2.3. Create the Employee table

2.3.1. From the Solution tree view, Select "Tables", and then click on "+ New" > "Table" > "Table":

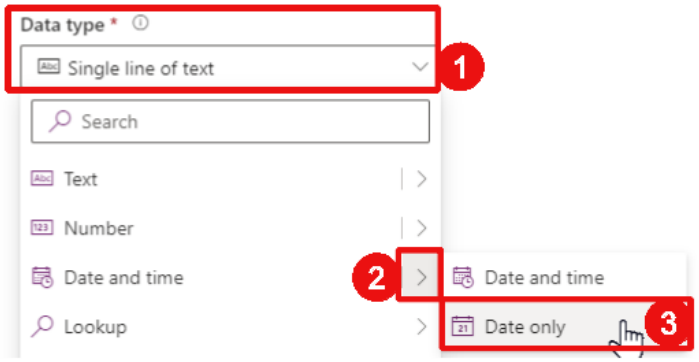


2.3.2. Create the table with the information below:

Display name: Employee
Plural name: Employees
Primary column Display name: Employee Name
Click "Save"

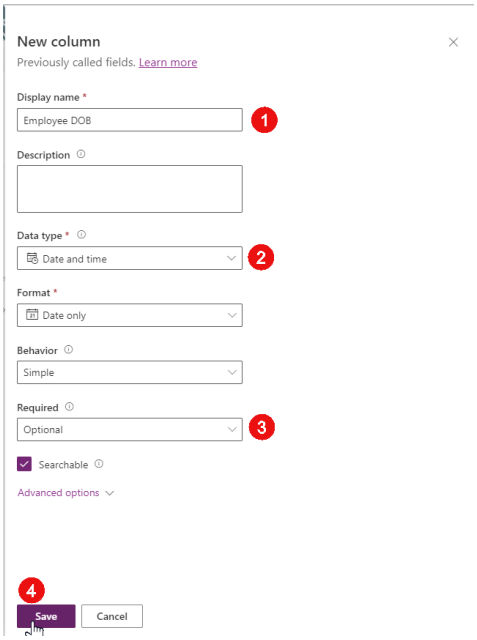
2.3.3. As for the Location table, open the Columns view and click "New column" to add the date of birth column:

Display name: Employee DOB
Data type: Date and time > Date only



Required: Optional

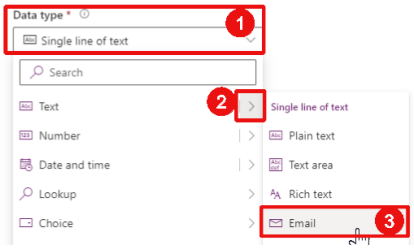
Click "Save":



2.3.4. Add another column for the email address:

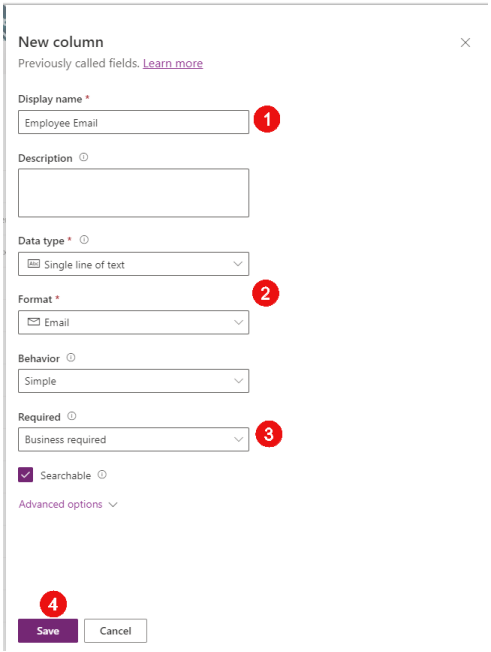
Display name: Employee Email

Data type: Text > Email

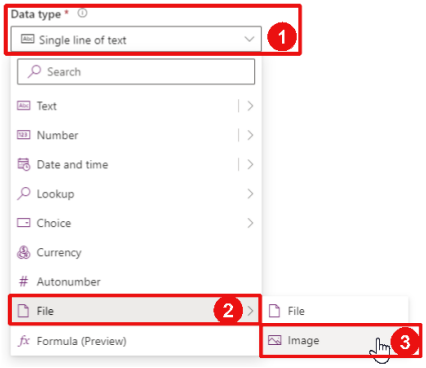


Required: Business required

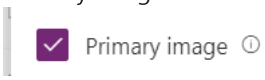
Click "Save":



2.3.5. Let's add a column for the picture of the employee:
Display name: Employee Picture
Data type: File > Image:



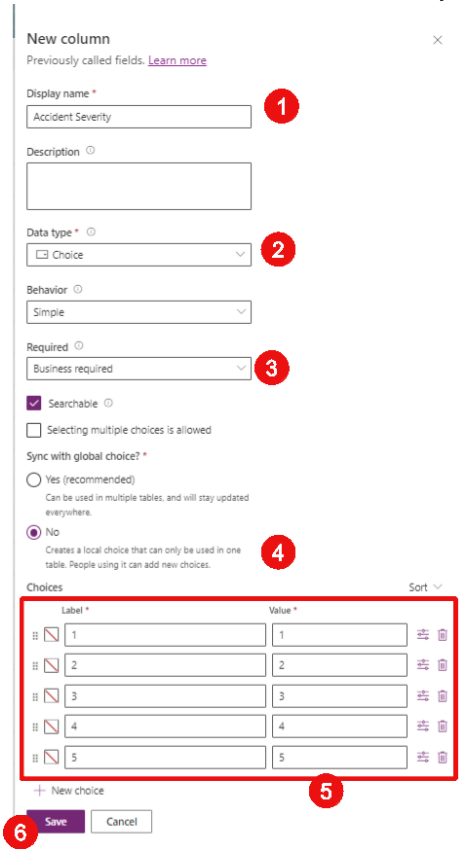
Pirmary image: Yes



2.4. Create the Accident Types table

2.4.1. As for the Location and Employee tables, create a new table as below:
Display name: Accident Type
Plural name: Accident Types
Primary column Display name: Type

2.4.2. Once saved, let's create an Accident Severity column as a choice. This time we will use a local choice because the severity concept is proper to the Accident Type table and shouldn't be confused with the severity concepts that we might use in other tables:

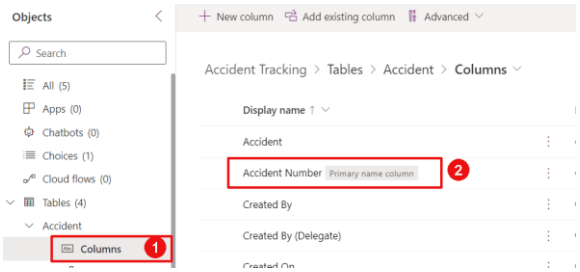


2.5. Create the Accident table

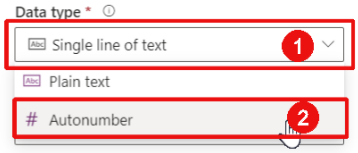
2.5.1. Create the Accident table as below:

Display name: Accident
Plural name: Accidents
Primary column Display name: Accident Number

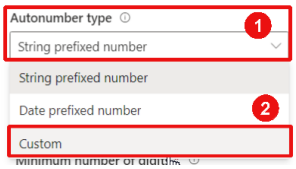
2.5.2. This time, the primary column is going to be an automatically calculated number so we will need to update the column. Open the column view and select the Accident Number column:



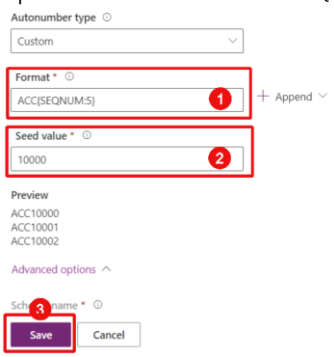
2.5.3. Update the Data type to "Autonumber":



2.5.4. Update the Autonumber type to "Custom":



2.5.5. Update the Format to "ACC{SEQNUM:5}", the Seed valu to "10000" and click "Save":



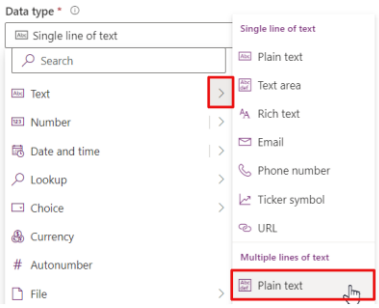
2.5.6. Add the Accident Description column:

Display name: Accident Description
Data type: Text > Single line of text > Plain text

Format: Text
Behaviour: Simple
Required: Business Required
Searchable: Yes

2.5.7. Add the Accident Date column:
Display name: Accident Date
Data type: Date and Time > Date and time
Format: Date and time
Behaviour: Simple
Required: Business Required
Searchable: Yes

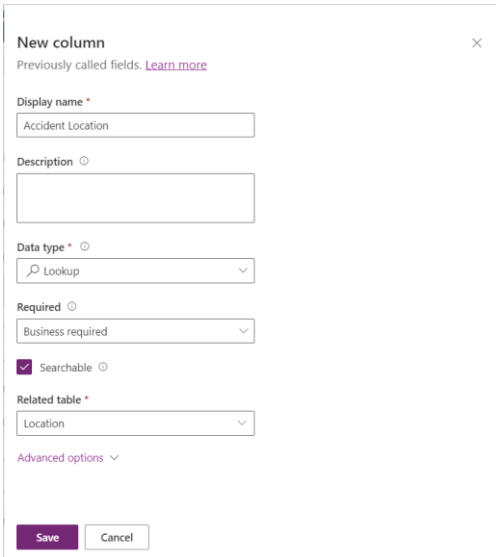
2.5.8. Add the Manager Comments column:
Display name: Manager Comments
Data type: Text > Multiple lines of text > Plain text



Format: Text
Required: Optional
Searchable: Yes

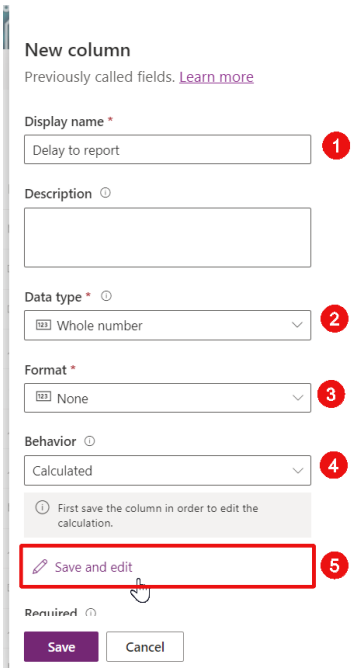
2.5.9. Add the Manager Reviewed column:
Display name: Manager Reviewed
Data type: Choice > Yes/no
Behaviour: Simple
Required: Optional
Searchable: Yes

2.5.10. Now let's add the lookup column to point to the Location table. Each Lookup table automatically creates a One-to-Many relationship between the two tables:
Display name: Accident Location
Data type: Lookup > Lookup
Required: Business Required
Searchable: Yes
Related table: Location

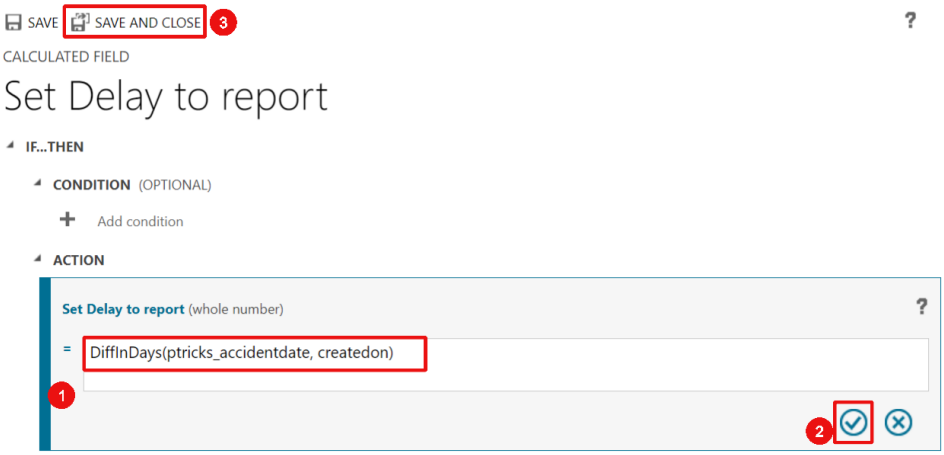


2.5.11. Do the same for the Accident Type:
Display name: Accident Type
Data type: Lookup > Lookup
Required: Business Required
Searchable: Yes
Related table: Accident Type

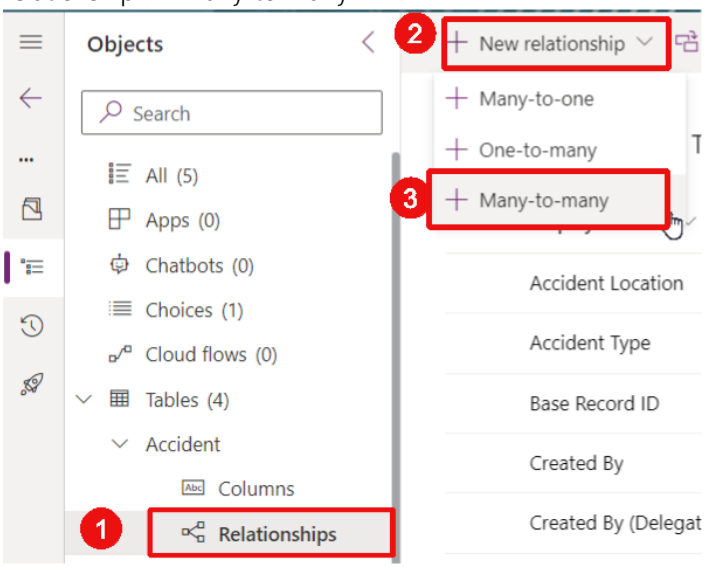
2.5.12. Now let's create a calculated column to show how many days passed between the accident date and the creation date in Dataverse:
Display name: Delay to report
Data type: Number > Whole number
Format: None
Behaviour: Calculated
Click "Save and Edit" to define the calculation rule:



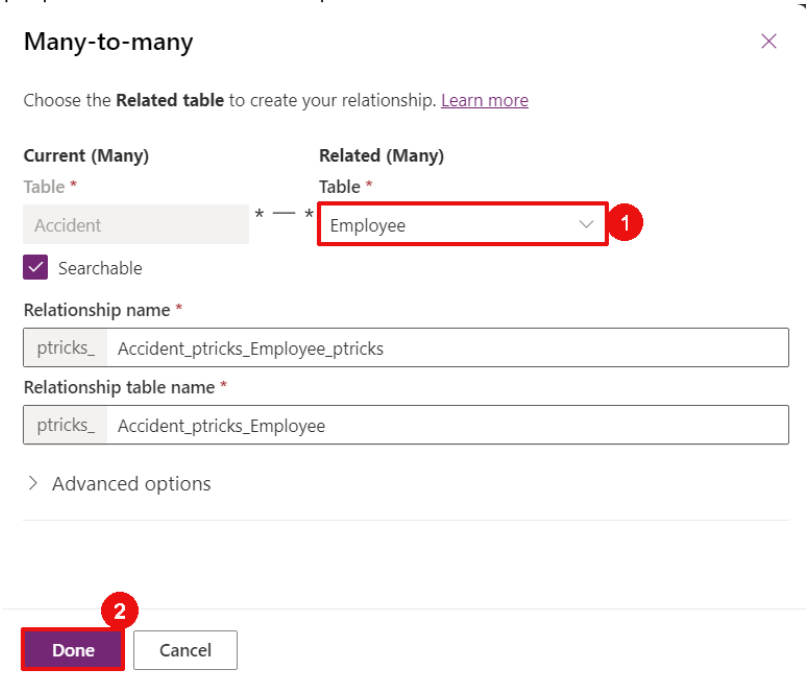
2.5.13. A new windows opens. Add an Action by click on "+ Add action" and add the formula below:
DIFFINDAYS(ptricks_accidentdate, createdon)
Then click the validation icon and finally "SAVE AND CLOSE":



2.5.14. We now need to add a many-to-many relationship between the Accident and the Employee tables. From the solution tree view (left-hand side), click on Tables > Accident > Relationships to open the relationships view of the Accident table. Then click "New relationship">"Many-to-many":



2.5.15. Select the Employee table as Related and click Done. Notice that a name is automatically proposed for the relationship as well as for the hidden intersect table:

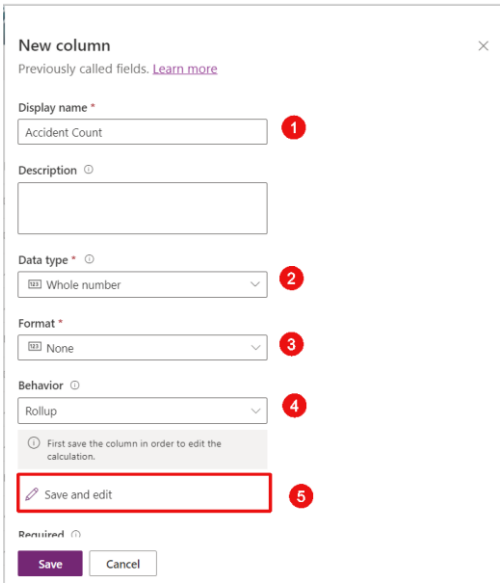


2.6. Add a roll up column in the Accident Type table

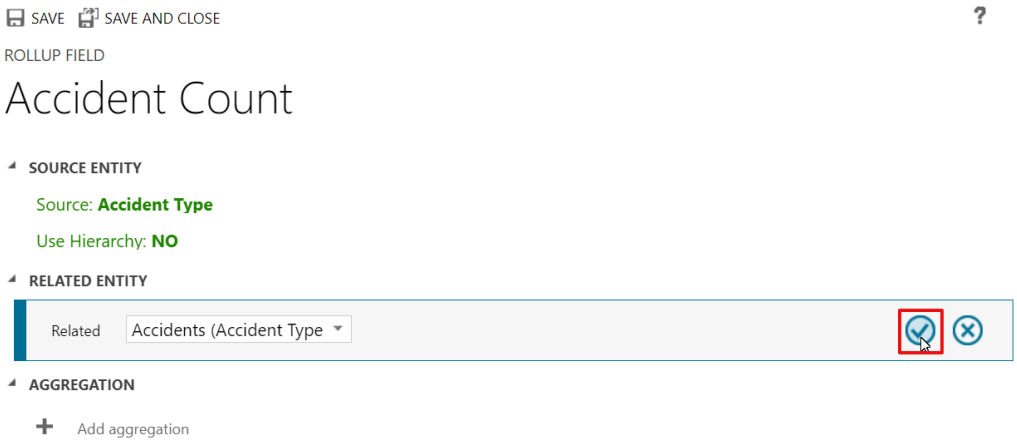
Now that the Accident and Accident Type tables are both created and related to each other (thanks to the look up column in the Accident table), we can add a roll up column in the Accident Type table. Roll up columns allow to do calculation on a related table. We will do this to count how many accidents occurred for each accident type.

2.6.1. Open the Accident Type table and open the Columns view

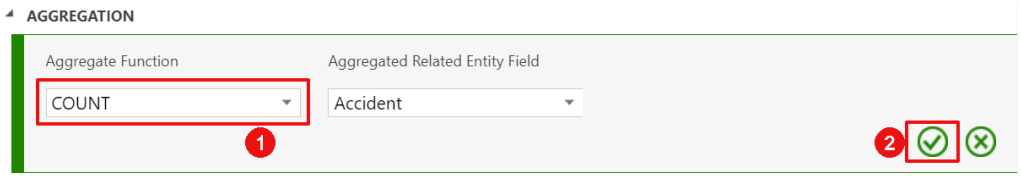
2.6.2. Add a new column with the details below:
Display name: Accident Count
Data type: Number > Whole number
Behaviour: Rollup
Format: None
Click "Save and edit":



2.6.3. A new window opens. Click "Add related entity" and validate the Accident table as the related entity:



2.6.4. Click "Add aggregation", select the COUNT function and validate the aggregation:



2.6.5. Click "SAVE AND CLOSE"

3. Create the Model Driven App

We created the data structure in the last Chapter. We will now be preparing the interface elements to render in the Model Driven App as needed with Views and Forms prior to composing the Model Driven App.

3.1. Configure the Views for the Location table

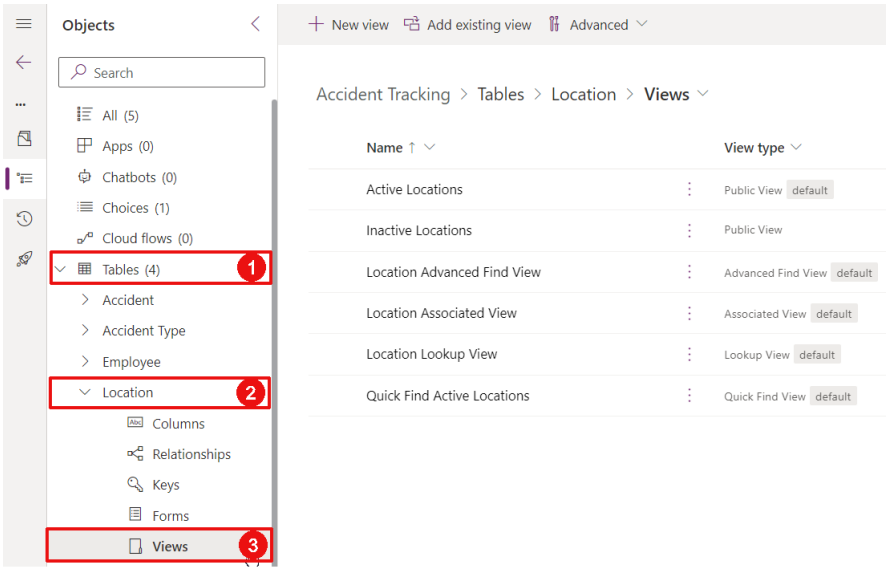
Views are the component allowing to display the table data. More information about views are available in this Microsoft learning module.

We will now edit the Location table views. Each work accident will be associated to a company location, and all locations are stored in the location table.

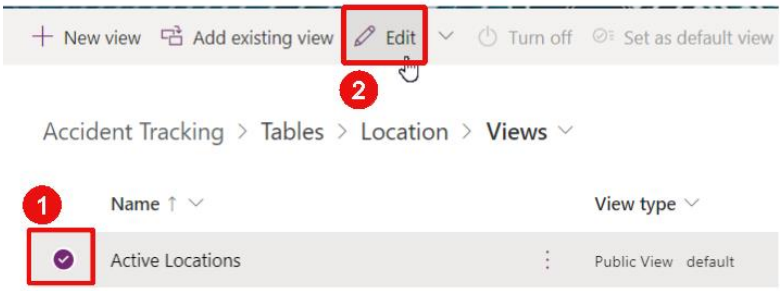
3.1.1. Go to https://make.powerapps.com/home and switch to the environment that you have available for this training.

3.1.2. On the left-hand panel, click on "Solutions" and open the "Accident Tracking" solution that we created in the previous Chapter

3.1.3. On the solution Objects tree view panel (on the left), extend the "Tables" > "Location" > and click on "Views":



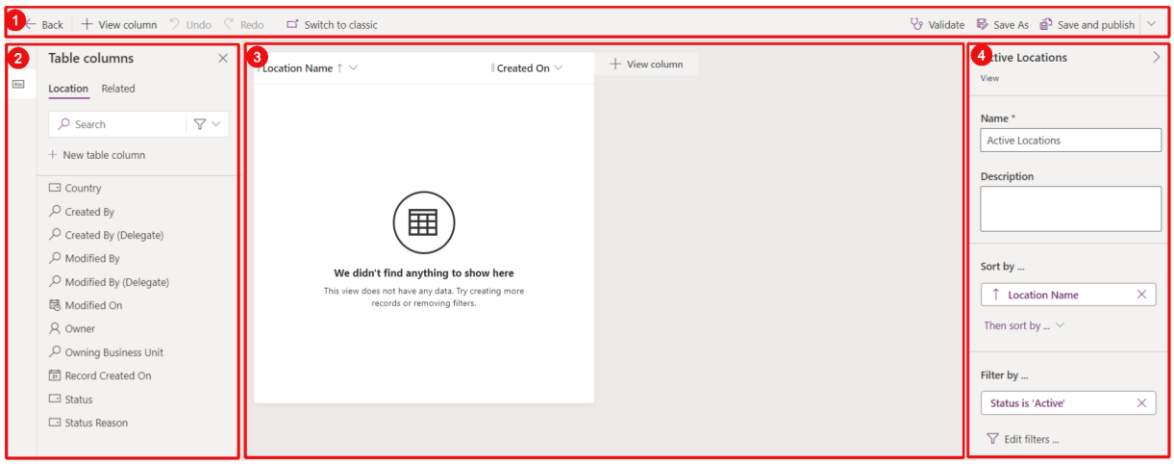
3.1.4. Select the "Active Locations" view and click "Edit":



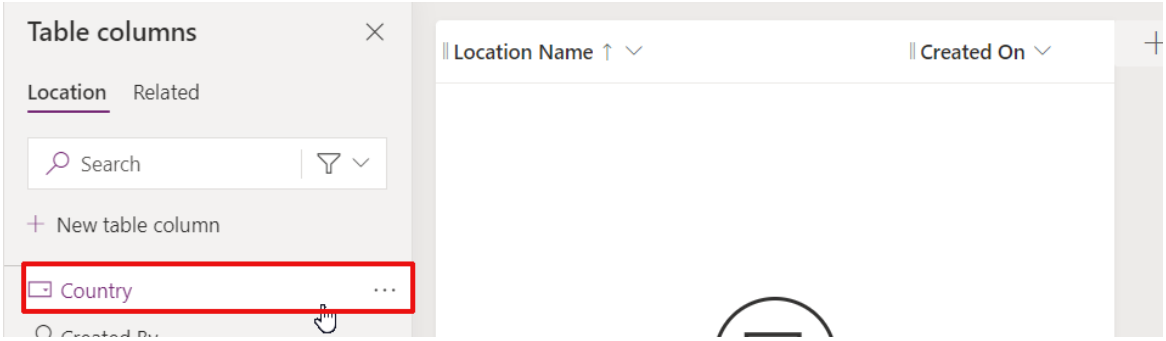
3.1.5. Notice that the screen is divided in 4 main sections:

1. The top toolbar allowing you to save and publish your view as well as perform other actions,
2. The Table columns panel on the left, showing you the different columns that you can add to the view,
3. The main canvas in the middle, where you can see the view you are editing,

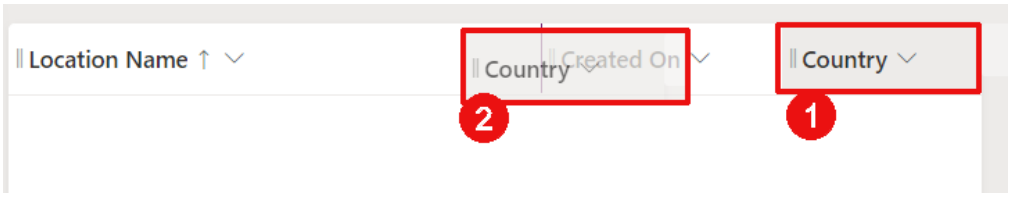
4. An additional configuration panel on the right, where you can adjust some settings.



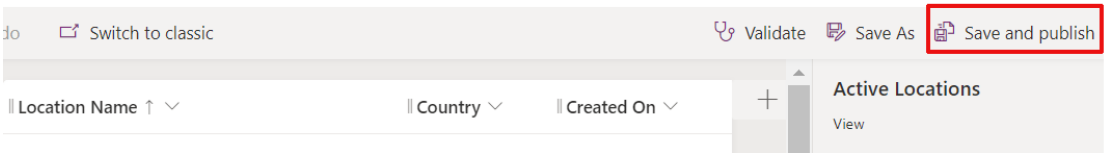
3.1.6. On the "Table columns" panel on the left, select the Country column to add it to the view. Notice that the column stops showing on the panel as soon as it is added to the view:



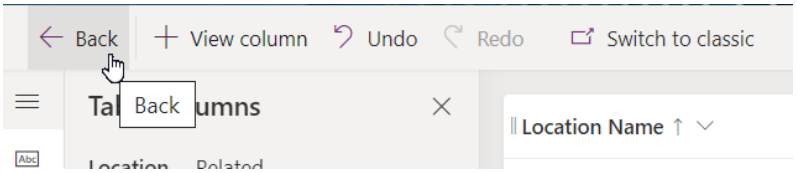
3.1.7. Adjust the order of the views with drag and drop to move the "Country" column before the "Created On" column:



3.1.8. Save and publish the View:

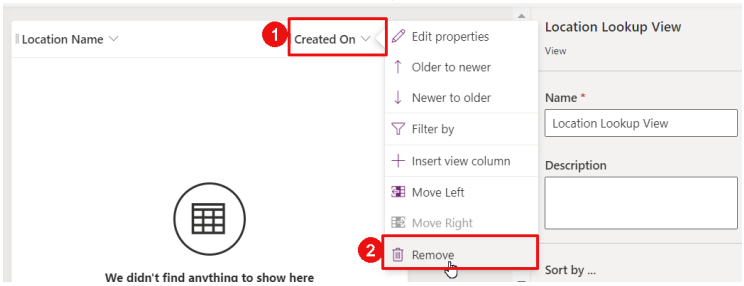


3.1.9. Click to "Back" to return to the Solution:



3.1.10. Let's now update the "Location Lookup View". This view is used when a Lookup column points to the Location table. For instance, it is used when creating an Accident and selecting the location. Select the "Location Lookup View" and click "Edit".

3.1.11. Remove the Created On column by clicking on it and select "remove":



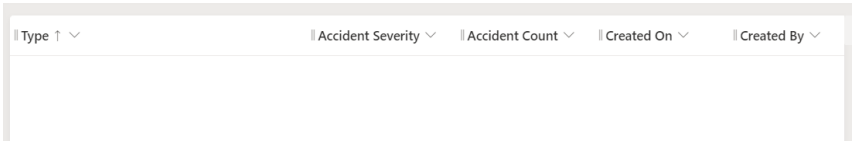
3.1.12. Add the Country column by selecting it from the left-hand panel

3.1.13. Save and Publish

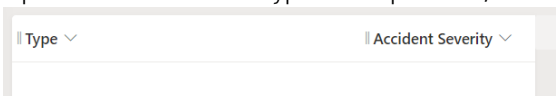
3.1.14. Click "Back" to return to the Solution

3.2. Configure the Views for the Accident table

3.2.1. Reproduce the same steps to update the "Active Accident Types" view. It should look as below:

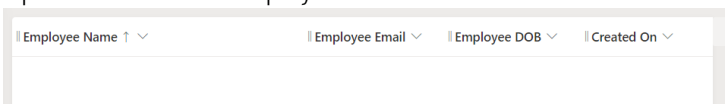


3.2.2. Update the "Accident Type Lookup View", it should look as below:

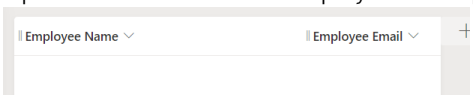


3.3. Configure the Views for the Employee table

3.3.1. Update the "Active Employees" view as below:

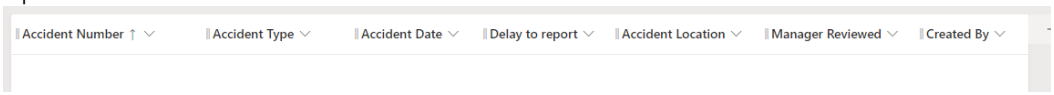


3.3.2. Update the view called "Employee Lookup View" as below:

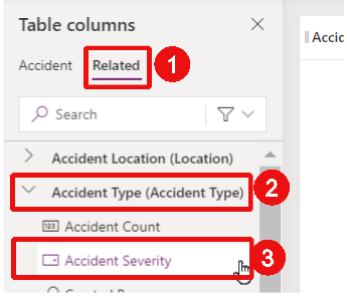


3.4. Configure the Views for the Accident table

3.4.1. Update the "Active Accidents" view as below:



3.4.2. We also would like to show some information from other tables. Let's add the Accident Severity column into that view. Select the "Related" tab on the left-hand panel. Extend the Accident Type table, and select "Accident Severity":



3.4.3. Reproduce the same steps to add the Country column from the Accident Location table

3.4.4. Adjust the column in the view to be in this order:

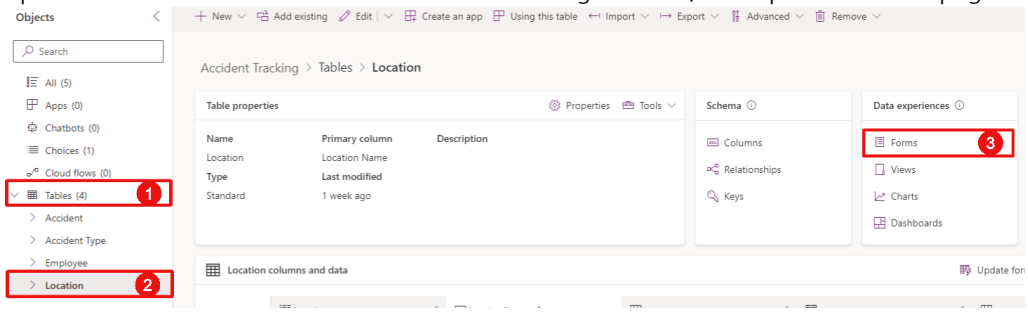
- Accident Number
- Accident Type
- Accident Severity (Accident Type)
- Accident Date
- Delay to report
- Accident Location
- Country (Accident Location)
- Manager Reviewed
- Created By

3.4.5. Save and publish, and then go back to the Solution

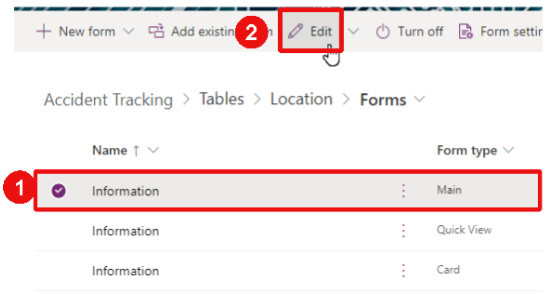
3.5. Configure the Forms for the Location table

Forms are the components allowing users to edit data. More information about forms can be found here.

3.5.1. Open the Location table from the Accident Tracking solution, and open the Forms page:

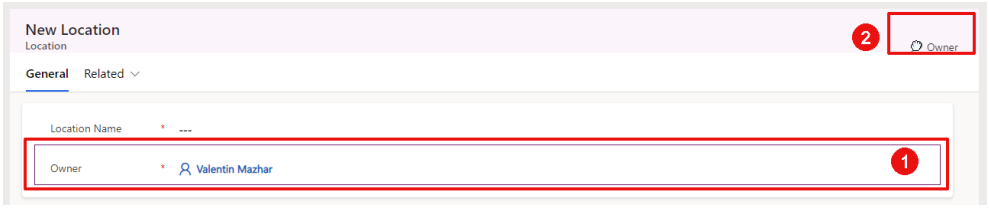


3.5.2. Select the Main form and click "Edit":

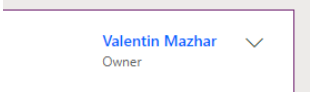


3.5.3. Notice that the layout is similar as for the views. The columns that can be added on the form are shown on the left-hand panel. The main section shows the actual form, and the right-hand panel shows some additional information and properties.

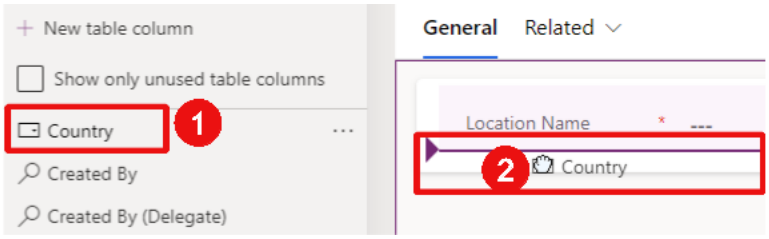
3.5.4. From the main section, drag and drop the "Owner" field to the top right-hand corner of the form:



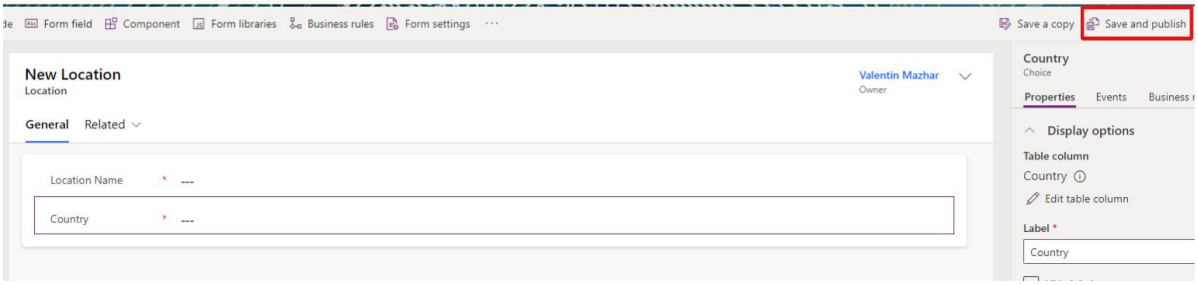
It should then look like this:



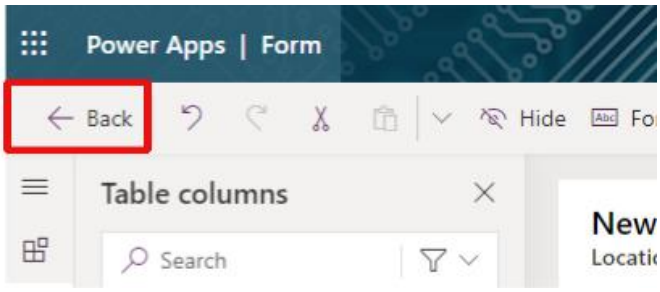
3.5.5. Drag and drop the Country column from the left-hand panel to the main section of the form, below the Location Name field:



3.5.6. Click ·Save and publish":



3.5.7. Click "Back" to return to the Solution

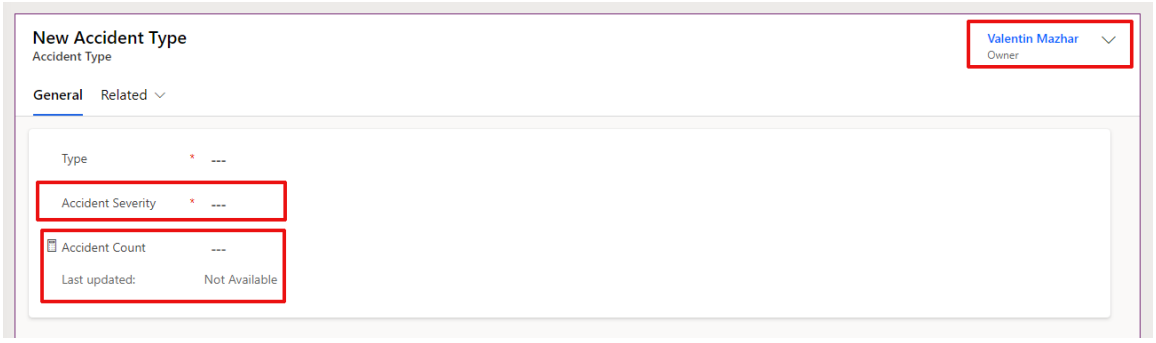


3.6. Configure the Forms for the Accident Type table

3.6.1. Reproduce the same steps to update the Main form of the Accident Type table and proceed to the below:

- Move the Owner field to the top
- Add the Accident Severity field
- Add the Accident Count field

The form should look this way:



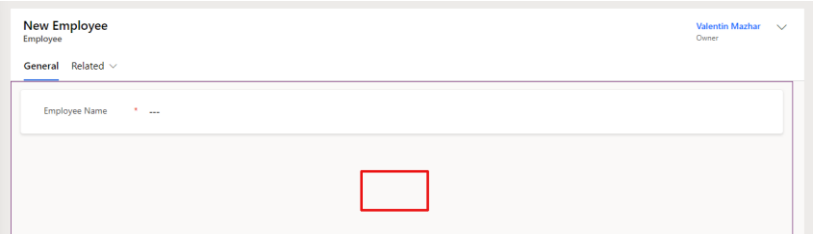
3.6.2. Save and publish, and return to the Solution by clicking "Back"

3.7. Configure the Forms for the Employee table

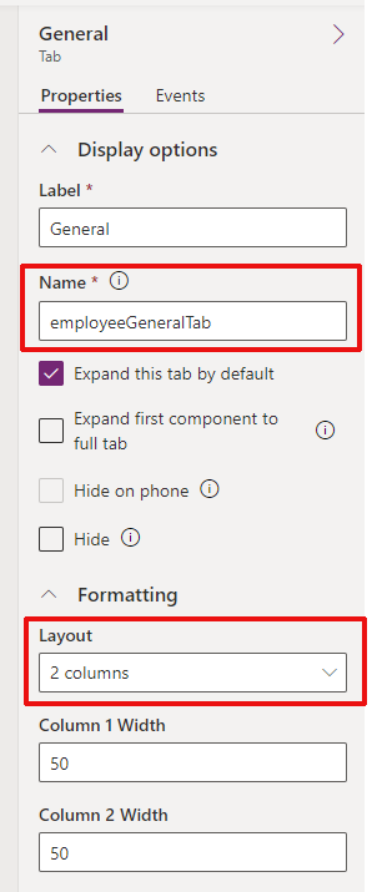
3.7.1. Select and edit the Main form of the Employee table

3.7.2. Move the Owner field to the top of the form

3.7.3. Click on the background of the form to ensure the right selection:

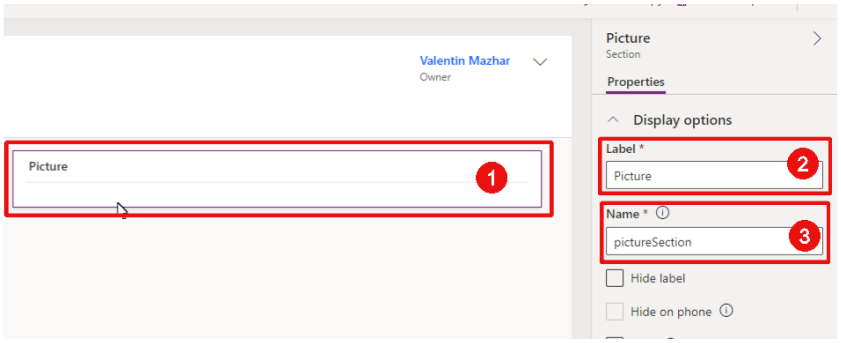


3.7.4. On the right-hand panel, add a name to the Tab and change the Layout to 2 columns:

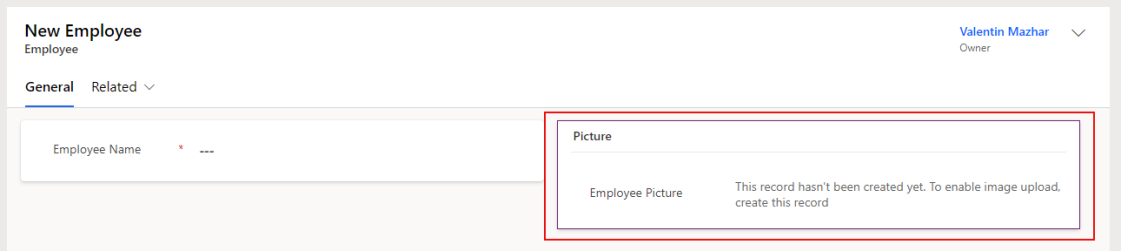


(If you do not see this panel with these properties, make sure to click on the background of the form to select the tab and not another component. You will Might also need to extend the "Formatting" subsection.)

3.7.5. Notice that the form is now divided in 2 sections. Select the New Section and update its name and Label as below:

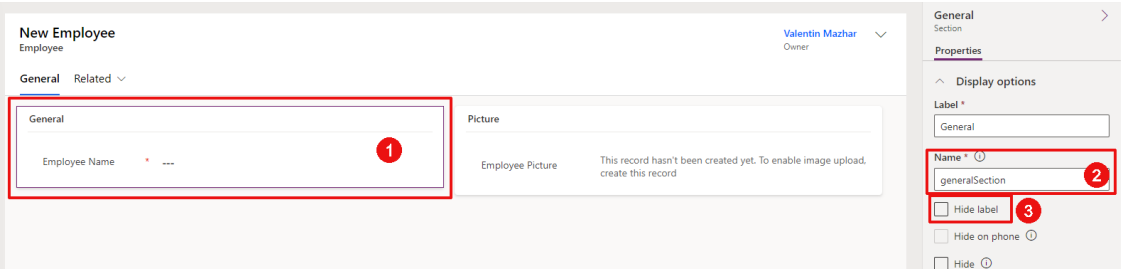


3.7.6. With the Picture section selected, add the Picture field to the form from the column panel on the left:



3.7.7. Update the properties of the other section as below:

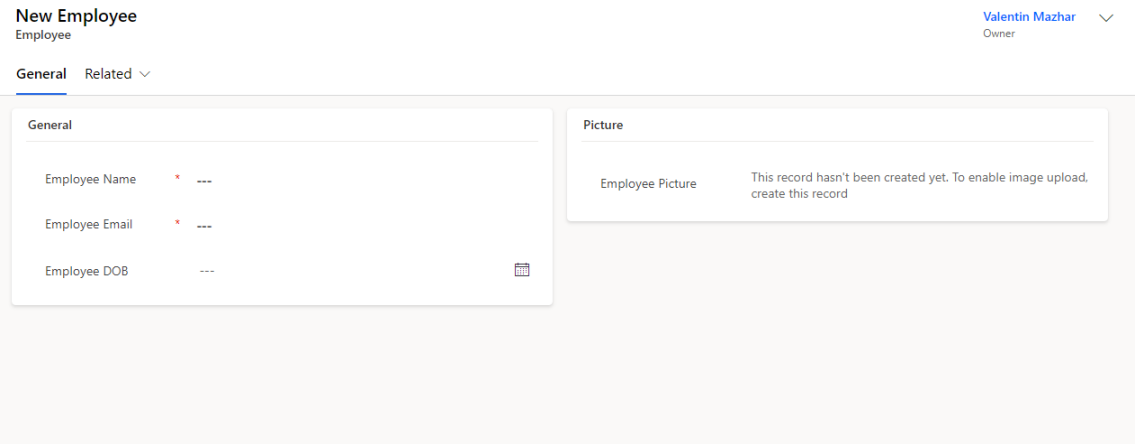
Select the other section
Update the Name to "generalSection"
Unhide the label



3.7.8. With the general section selected, add the fields below:

Employee Email
Employee DOB

3.7.9. Confirm that the form looks as below, make the necessary adjustments if not:



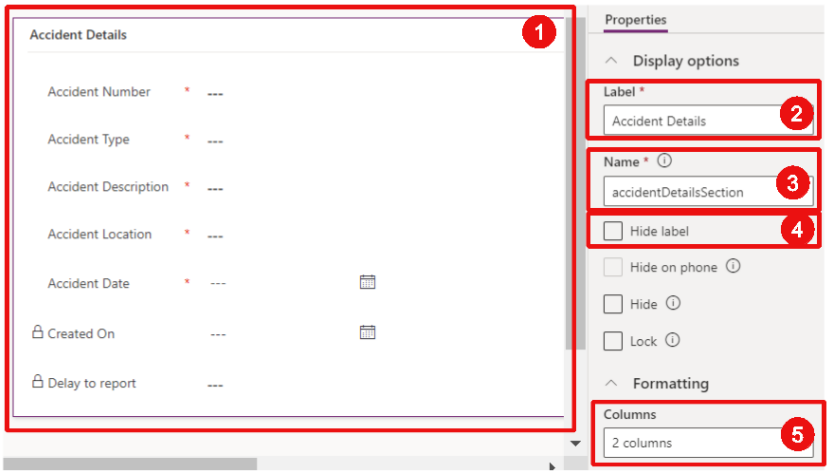
3.7.10. Save and publish, and return to the Solution

3.8. Configure the Forms for the Accident table

3.8.1. Select and edit the Main form of the Accident table
3.8.2. Move the Owner field to the top
3.8.3. Add the fields below to the form:

Accident Type
Accident Description
Accident Location
Accident Date
Created On
Delay to report

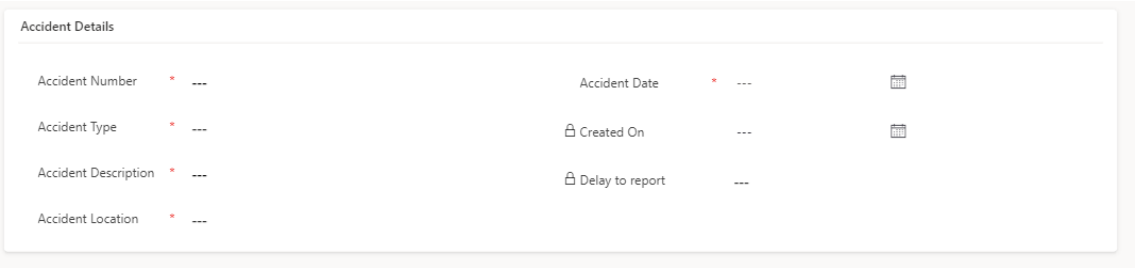
3.8.4. Select the main section and update the properties on the right-hand panel as below:



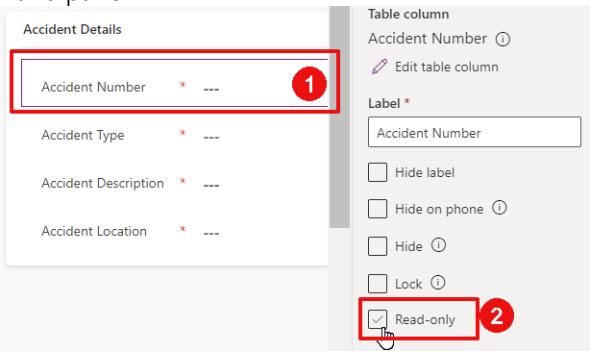
3.8.5. Drag and drop the fields below to the second column of the section:

Accident Date
Created On
Delay to report

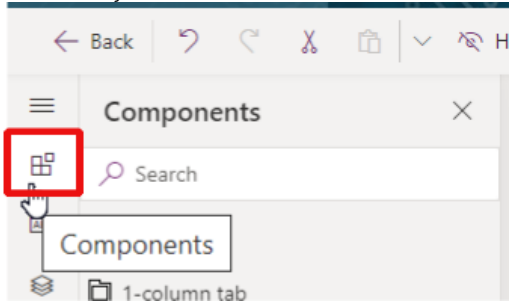
Confirm that the form looks like the screenshot below:



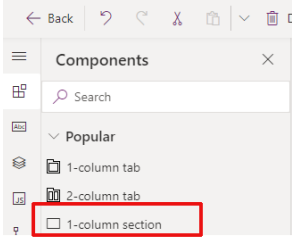
3.8.6. We want to make sure that the Accident Number field cannot be manually edited by a user since it should be automatically defined. Select the field and tick "Read-only" from the right-hand panel:



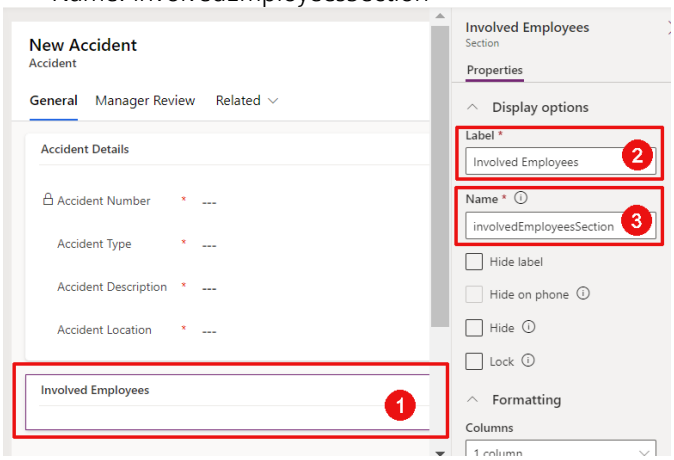
3.8.7. On the very left, from the vertical bar, click on the Components tab:



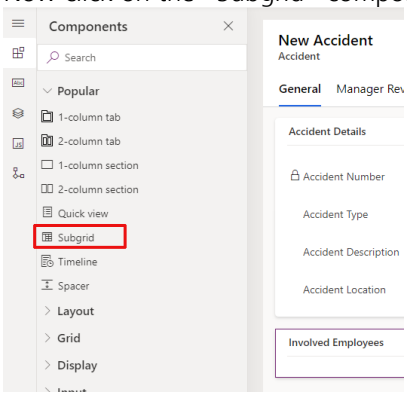
3.8.8. From the left-hand panel, click on "1-column section" to add a new section:



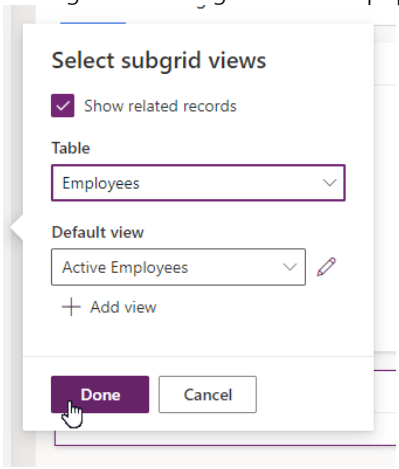
3.8.9. Select the section and update the right-hand properties of that section as below:
Label: Involved Employees
Name: involvedEmployeesSection



3.8.10. Now click on the "Subgrid" component to show all the related Employees:

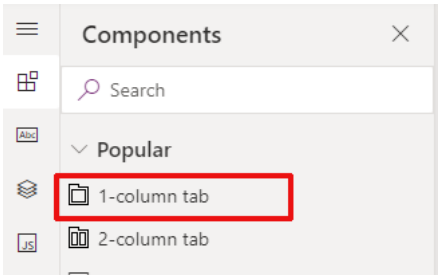


3.8.11. Configure the Subgrid with the popup as below:

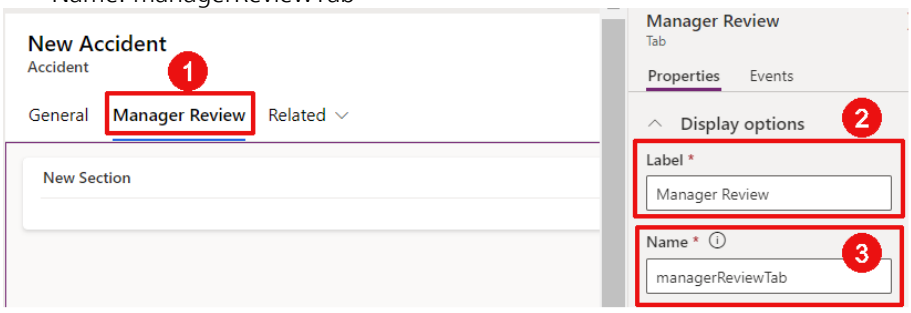


3.8.12. Hide the Subgrid label by ticking "Hide label" on the right-hand property panel, and rename it anyway as "Involved Employees".

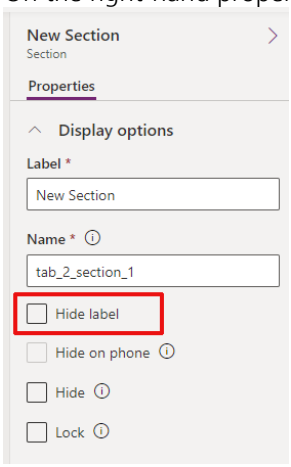
3.8.13. Let's now add a new tab. Select "1-column tab" to add a new tab to the form:



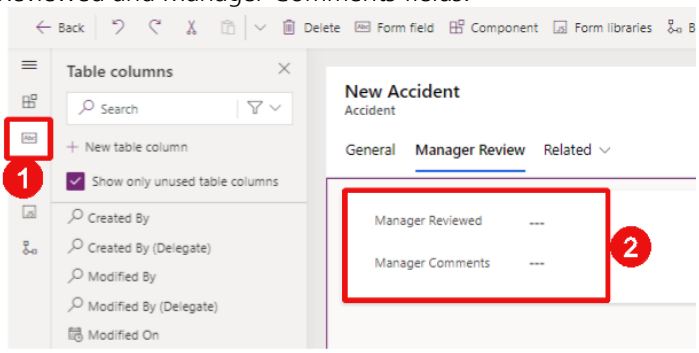
3.8.14. Select the new tab and update the right-hand properties as below:
Label: Manager Review
Name: managerReviewTab



3.8.15. Select the section "New Section" on the canvas.
(if you do not see the section, click on the new "Manager Review" tab to refresh the canvas)
On the right-hand property panel, tick "Hide label":



3.8.16. On the left-hand panel, switch back to the column views and add the columns Manger Reviewed and Manager Comments fields:



3.8.17. Save and publish the form, but do not go back to the main form

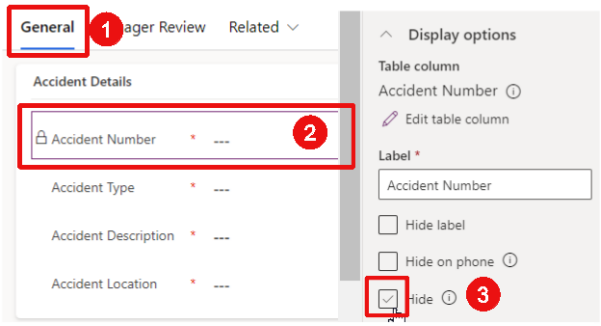
3.9. Configure the Business Rules for the Accident table

During the creation of an accident, the "Accident Number" field will be empty because the accident won't be saved into Dataverse yet. It is only after being saved that the Accident Number will be defined.

From a user perspective, seeing a mandatory and empty field that is read-only and cannot be updated might be confusing during the accident creation. Ideally, we would like this field to be hidden while the Accident is not created in Dataverse, and show it on the form as soon as the value has been automatically set after the creation.

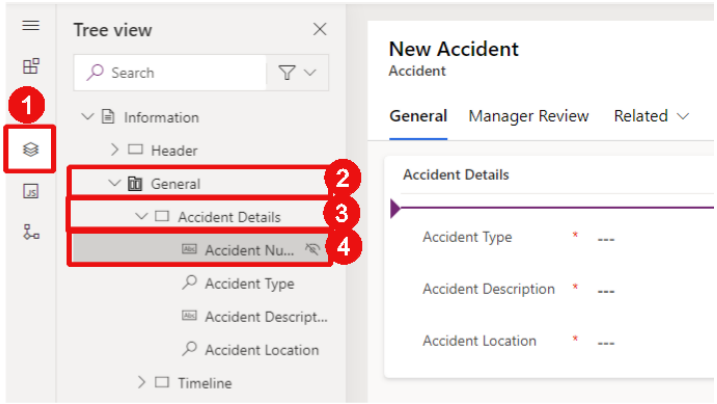
This can be achieved with a Business rule.

3.9.1. From the Accident Main form and the General tab, select the Accident Number field and tick "Hide" on the right-hand property panel:

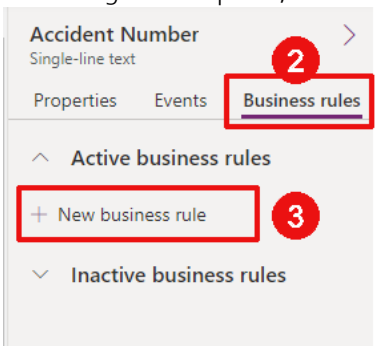


Now the field will always be hidden on the form.

3.9.2. Let's now create a Business Rule to automatically show the field on the form once the field is automatically populated. On the very-left-hand vertical bar, select the tree view tab and select the Accident Number field:



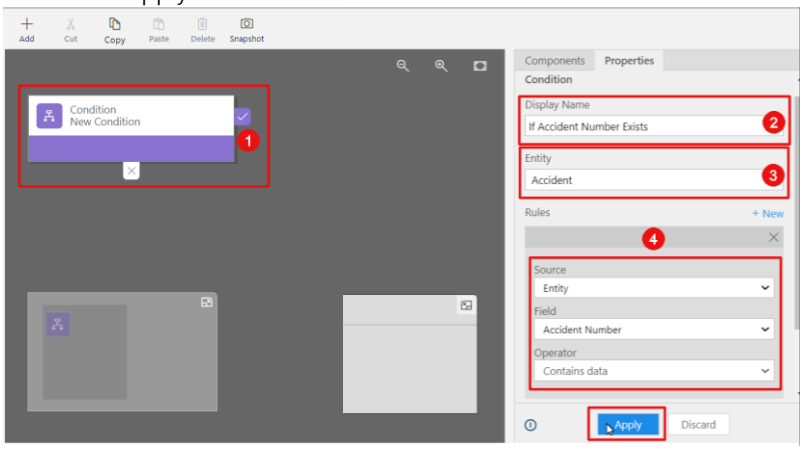
3.9.3. On the right-hand panel, select the Business rules tab and click on "New business rule":



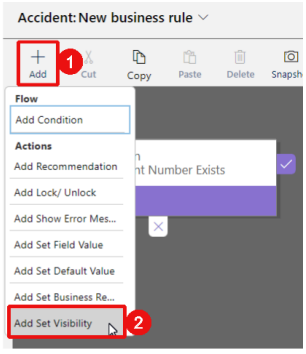
3.9.4. The business rule editor opens on a new browser tab. Select the Condition, add the display name "If Accident Number Exists" and configure the rule as below:

Source: Entity
Field: Accident Number
Operator: Contains data

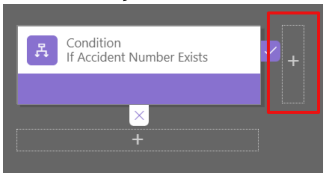
Then click Apply:



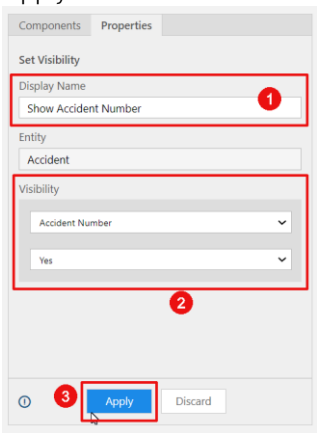
3.9.5. Click Add > Add Set Visibility to add a new action:



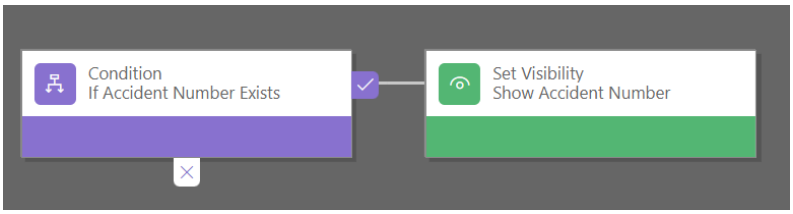
3.9.6. Select the yes branch of the condition where you want the action to be executed:



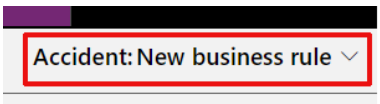
3.9.7. Add the display Name "Show Accident Number", set the visibility as below and click Apply:



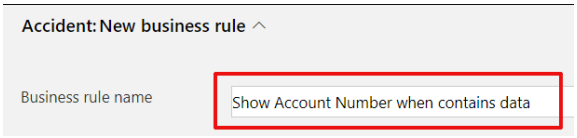
3.9.8. Confirm that the business rule looks like this:



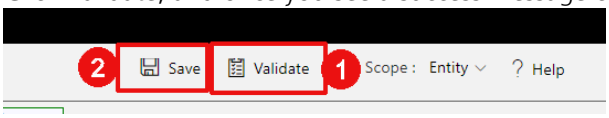
3.9.9. Let's add also a name to our business rule. On the top of the screen, extend "Accident: New business rule":



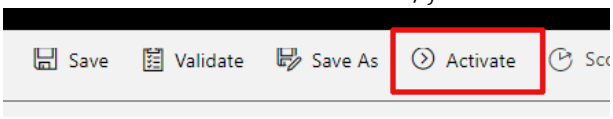
3.9.10. Add the name "Show Account Number when contains data":



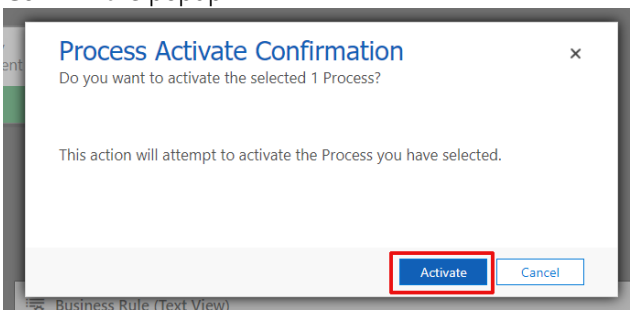
3.9.11. Click Validate, and once you see a success message click "Save":



3.9.12. Once the rule is validated and saved, you can Activate it:

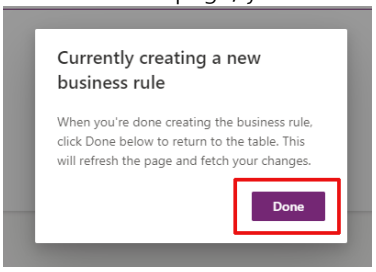


3.9.13. Confirm the popup:



3.9.14. Once activated, close the browser tab and return to the form tab

3.9.15. From the form page, you should see a new popup, click Done:



3.9.16. Note that the situation will be similar for two other fields: "Created On" and "Delay to report". Reproduce the steps 1 to 15 of this section:

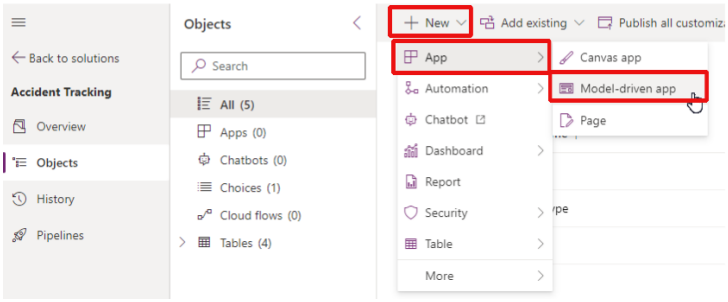
Hide the Created On field from the form
Create a Business Rule for the Created On field to set it visible when it contains data
Call this rule "Show Created On when contains data" and activate it
Do the same steps for the "Delay to report" field

3.9.17. Save and publish the form

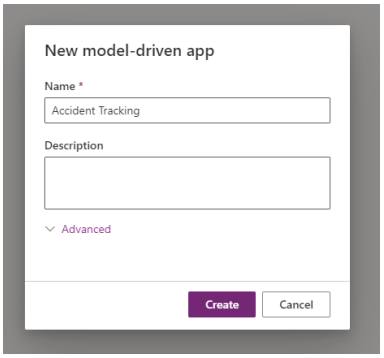
3.10. Start Composing the Model Driven App

We now have all the components we need to be able to start creating our Model Driven App.

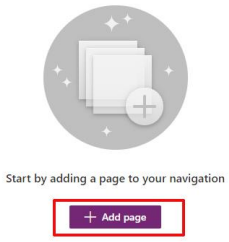
3.10.1. From the Solution page, click New > App > Model-driven app:



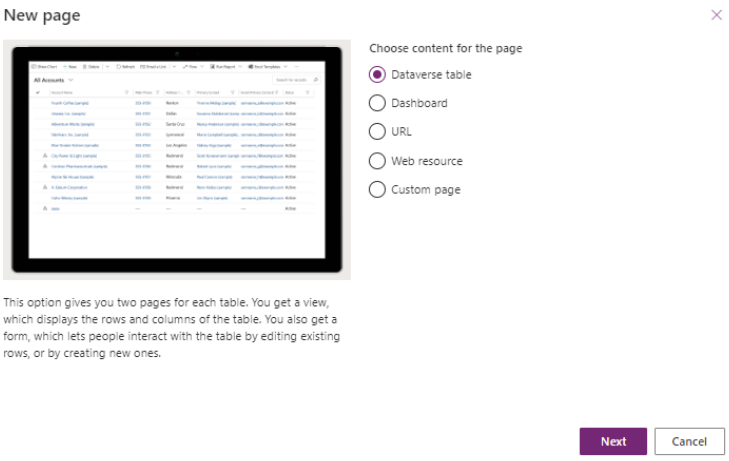
3.10.2. Give it the name: "Accident Tracking" and click Create:



3.10.3. Click "Add page":



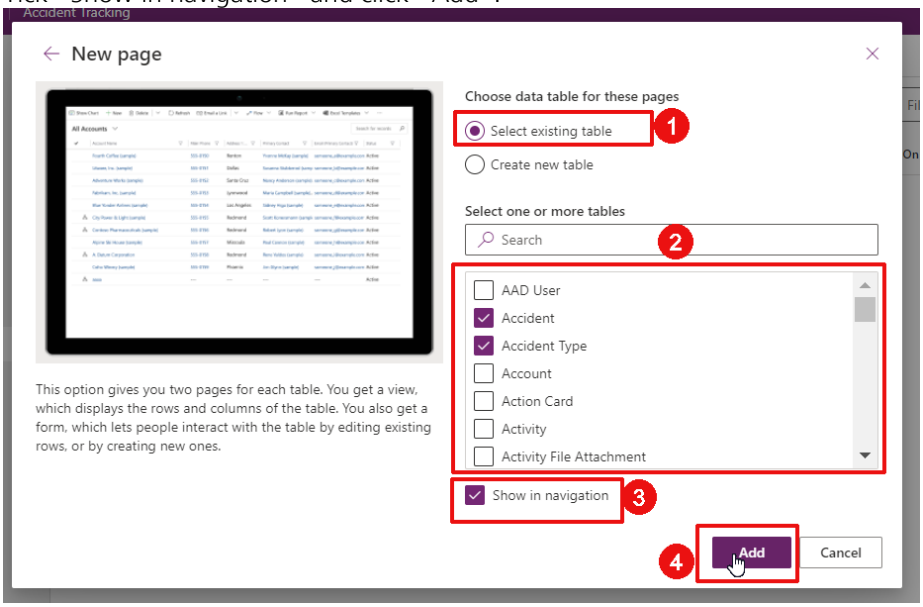
3.10.4. Select "Dataverse table" and click "Next":



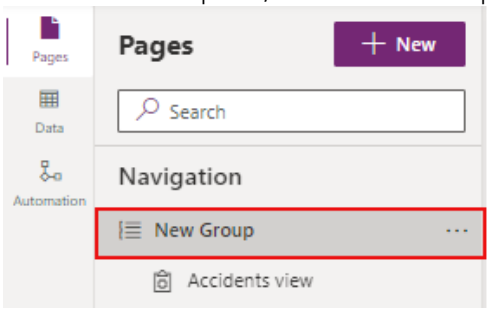
3.10.5. Select "existing table" and select the tables below:

Accident
Accident Type
Location
Employees

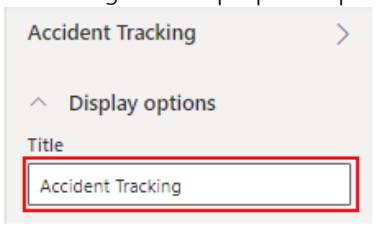
Tick "Show in navigation" and click "Add":



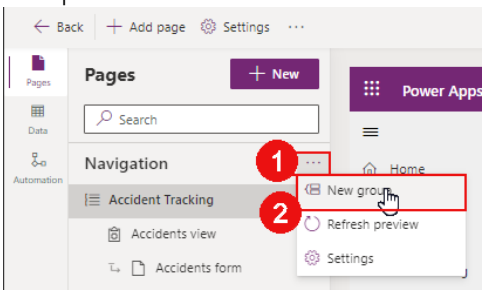
3.10.6. On the left-hand panel, select "New Group":



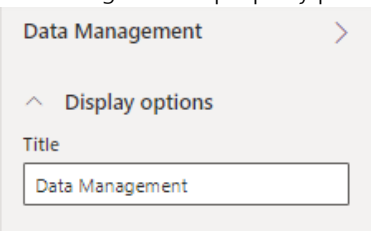
3.10.7. On the right-hand properties panel, rename the group into "Accident Tracking":



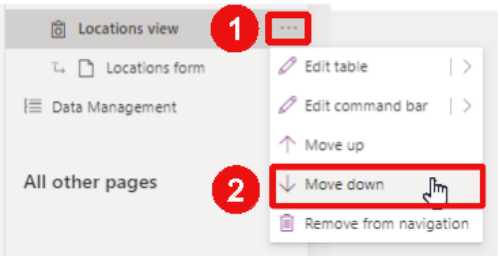
3.10.8. On the left-hand panel, click on the three dots next to "Navigation" and click "New Group":



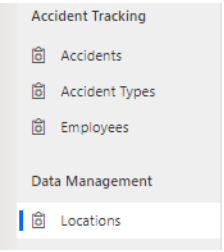
3.10.9. On the right-hand property panel, rename it into "Data Management":



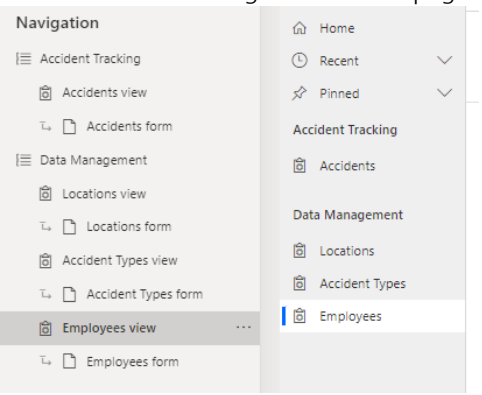
3.10.10. On the left-hand panel, click on the three dots next to "Location view" and click "Move down:



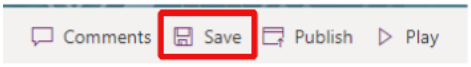
Notice that the tab is now in the Data Management group:



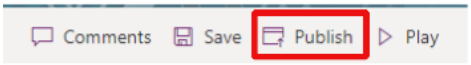
3.10.11. Do the same things to have the pages organized as below:



3.10.12. Click Save from the top command bar:



3.10.13. Click Publish:



3.10.14. Click Play to test the App in another Tab:

3.10.15. As done during the first Chapter, explore the different pages and proceed to the below:

Create a few Location (2-3)
Create a few Accident Types (2-3)
Create a few Employees (2-3)

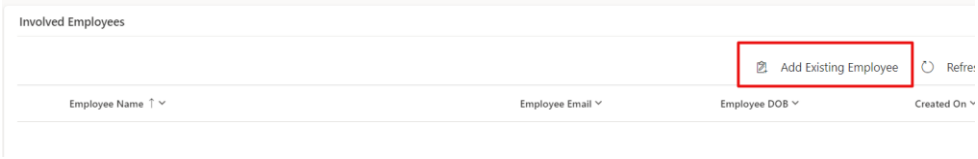
Notice the views and forms as you proceed

3.10.16. Create an Accident

Notice that the Accident Number, Created On and Delay to report are not shown while you add the information

After saving the information, notice that these fields are appearing thanks to the created business rules

Saving also allows you to add the involved employees by clicking "Add Existing Employee":



4. Extend the Model Driven Apps

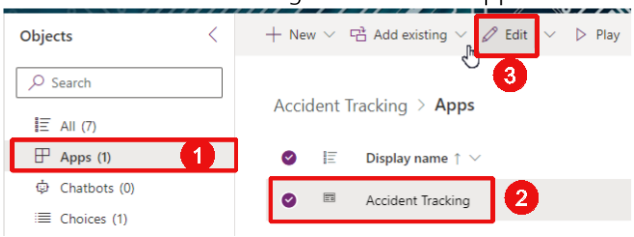
Model Driven Apps offer many more functionalities that what has been done so far. Let's extend it with additional features.

4.1. Create the custom page to add new Employees

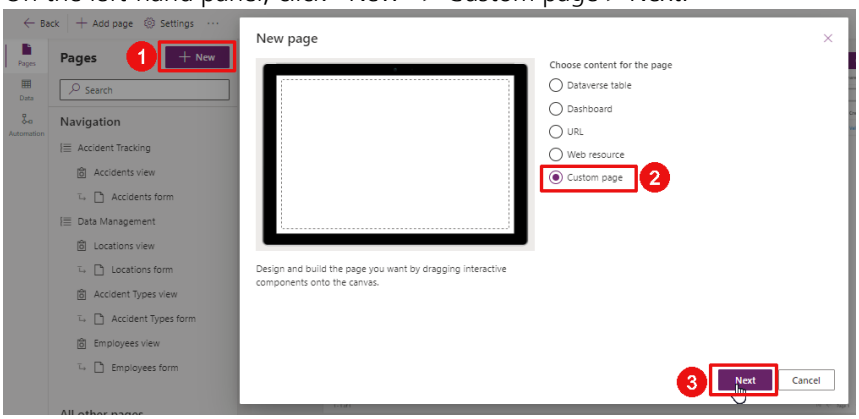
The current process to add Employees is taking too much time. Let's see how we could leverage Canvas Apps capability to create a people picker functionality to select employees from our user repertory in a faster way (Azure Active Directory).

In fact, if this was a real-case scenario, there would be ways to leverage other tables such as the systemuser table or the Microsoft Entra ID table to not even have to import employees manually and having this automatically synced. However… 1 step at a time 😊

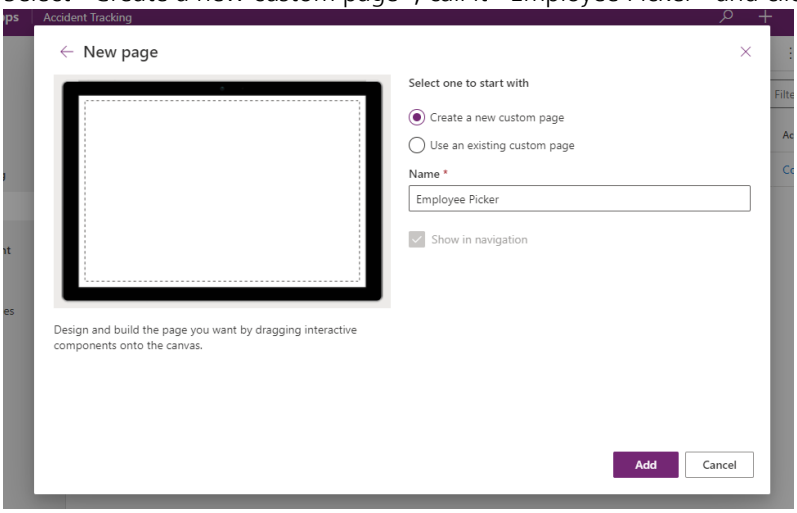
4.1.1. If you have closed the Model Driven App editor, go to your Accident Tracking Solution, find the Accident Tracking Model Driven app and click Edit:



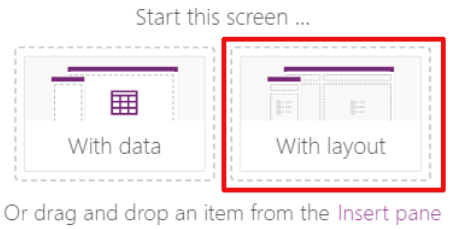
4.1.2. On the left-hand panel, click "New" > Custom page > Next:



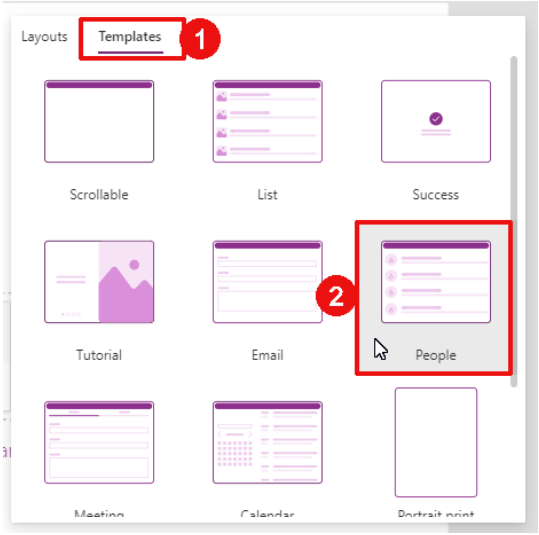
4.1.3. Select "Create a new custom page", call it "Employee Picker" and click "Add":



4.1.4. A new browser tab is opened. It looks the same way as to create a Canvas App. Select "With layout" to select an existing template:

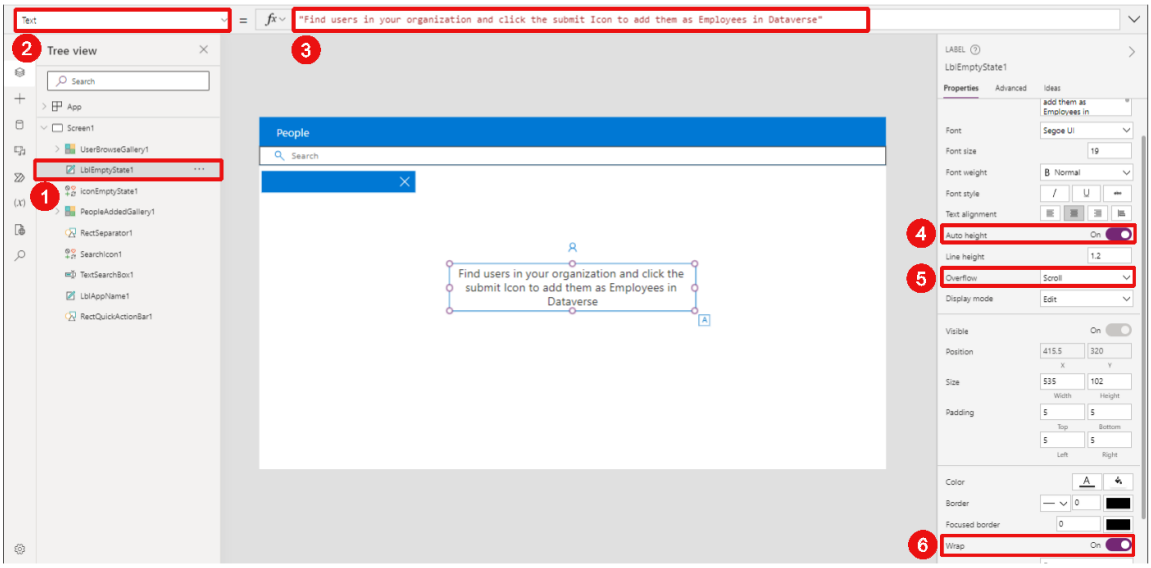


4.1.5. Select the Templates tab and click on the People template:

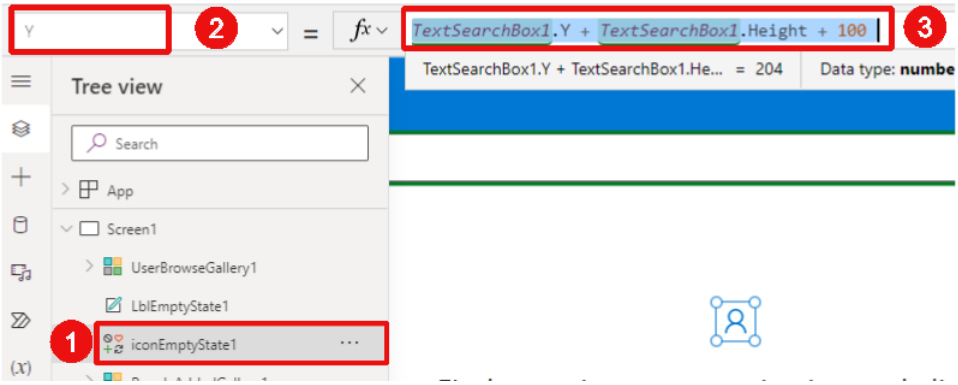


4.1.6. Let's slightly update the App to fit our need:

Update the Text property of the label "LblEmptyState1" to the below:
"Find users in your organization and click the submit Icon to add them as Employees in Dataverse"
Turn on "Auto height", set the "Overflow" property to "Scroll" and turn on the "Wrap" setting

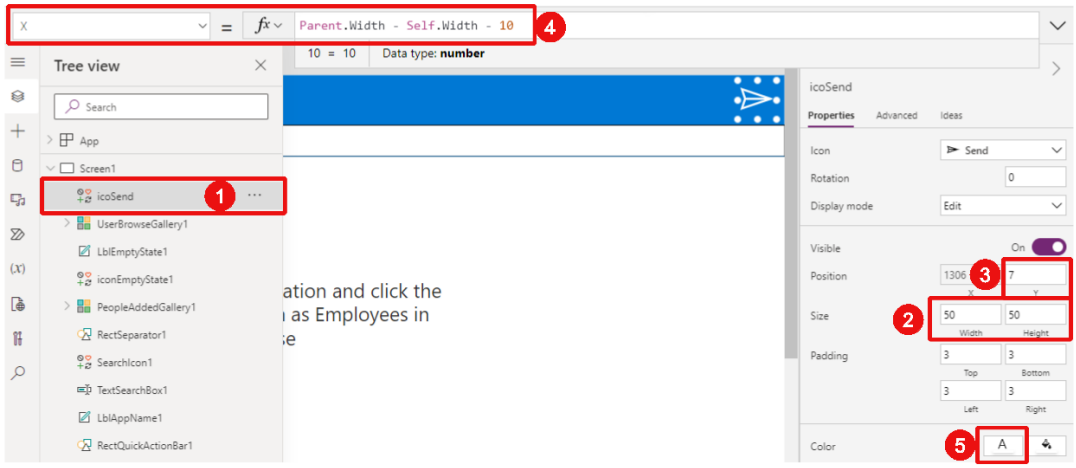


4.1.7. Select the icon "iconEmptyState1" and update its "Y" property to the below:
TextSearchBox1.Y + TextSearchBox1.Height + 100

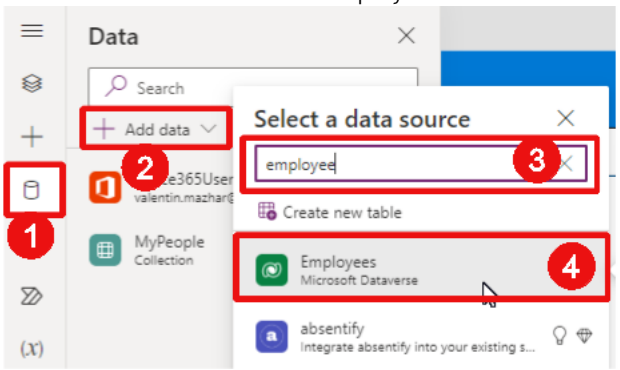


4.1.8. Insert a "Send" icon by clicking "Insert" form the top command bar > Icons > Send. Rename it into "icoSend":

4.1.9. Change the properties of this icon as below:
Width: 50
Height: 50
Y: 7
X: Parent.Width – Self.Width – 10
Color: Color.White
DisplayMode: Edit

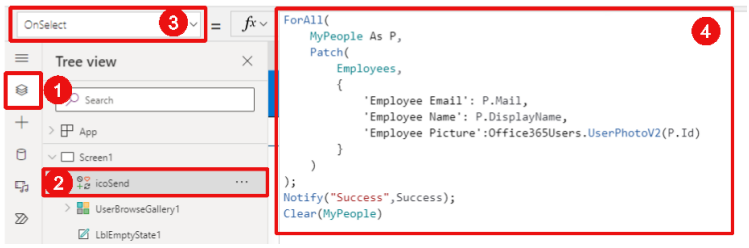


4.1.10. Let's add a connection to our Employees table. Click the Data tab on the left, then click Add data and search for the Employees table:



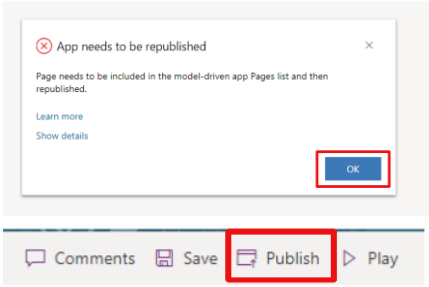
4.1.11. Go back to the Tree view and update the OnSelect property of the send icon as below:

```
ForAll(
    MyPeople As P,
    Patch(
        Employees,
        {
            'Employee Email': P.Mail,
            'Employee Name': P.DisplayName,
            'Employee Picture':Office365Users.UserPhotoV2(P.Id)
        }
    )
);
Notify("Success",NotificationType.Success);
Clear(MyPeople)
```



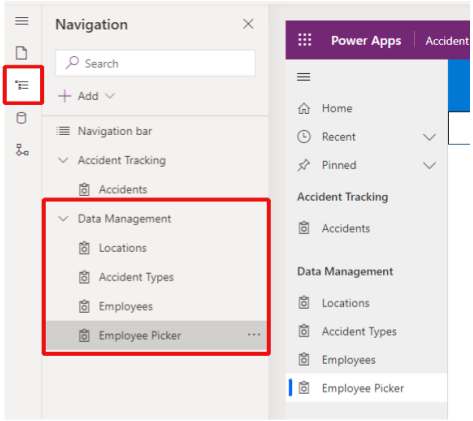
4.1.12. Save and publish the custom page:

4.1.13. Close the Browser tab and go back to the Model Driven App editing page. A popup indicates that the App needs to be republished, click ok and re-publish the App:

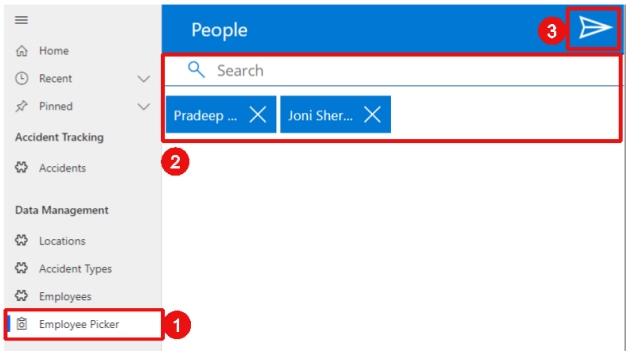


4.1.14. If there is a prompt to allow connections, proceed to allowing them

4.1.15. Notice that the Employee Picker page has been added in the App. If it is not already in the right place, go back to the navigation tab to move the custom page to the Data Management Group as below:



4.1.16. Save and publish the App and then test the functionality by clicking the Play button. Search for a few colleagues of yours and add them to the Employees table:



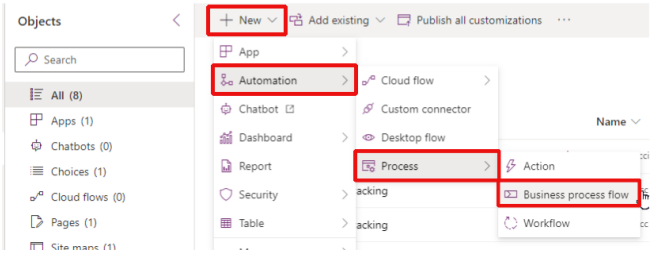
4.1.17. Open the Employees table and note the new employees added. Open one of them and note that all fields were populated except the Date of birth which is not available with the connector used in the Custom Page.

4.2. Create the Business Process Flow to create Accidents

In our use case, the creation process of an Accident is simple. However, imagine that you create a Claims management system. Once the initial details of a Claim are entered, multiple stakeholders might need to be involved at different stages to make some edits and validation. It is key that the right information is added in the right order by the right person.

In many situations it would be interesting to be able to guide the users of the App throughout such a process. This is what Business Process Flows are for. They offer a visual way to represent a process and guide the users through each step. Let's see how to create one for our Accident Tracking App.

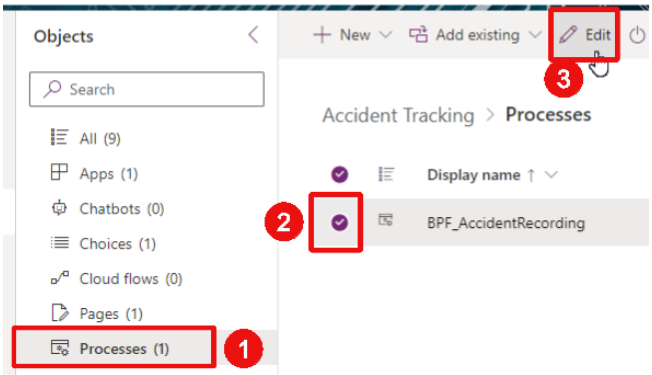
4.2.1. Open the Accident Tracking Solution, click New > Automation > Process > Business process flow:



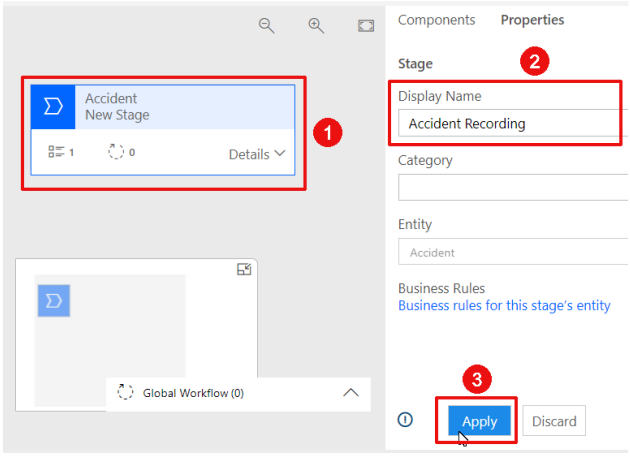
4.2.2. On the right-hand panel that opens on the screen, add the details below:

Display name: BPF_AccidentReporting
Name: ptricks_bpf_accidentrecording (automatic)
Table: Accident

4.2.3. A Business Process Editor should open in a new tab. If it does not, select the Process in the Solution and click Edit:



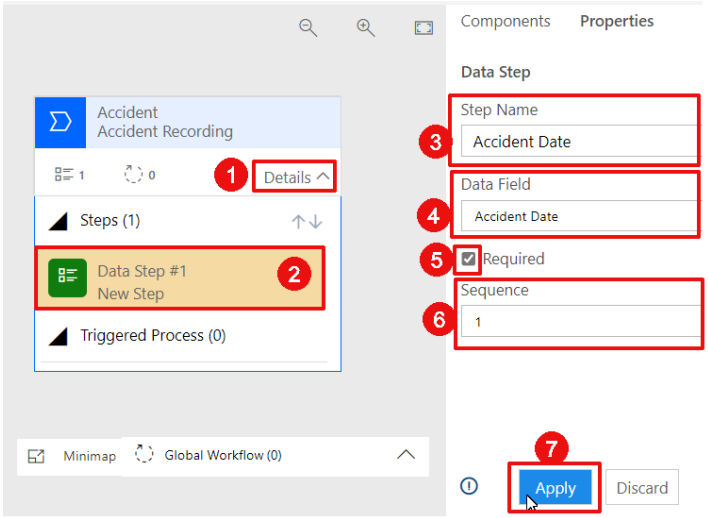
4.2.4. The interface looks similar as for business rules. Select the stage already on the screen and add "Accident Recording" as a Display Name, then click "Apply":



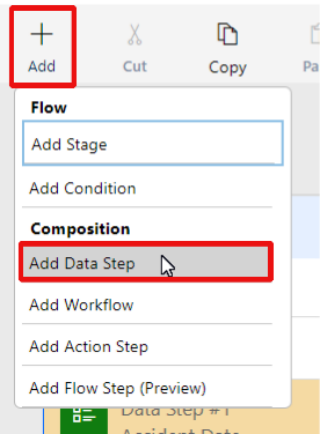
4.2.5. Extend the Details of the stage and select the first Data Step already there. Configure it as below:

Step Name: Accident Date
Data Field: Accident Date
Required: yes
Sequence 1

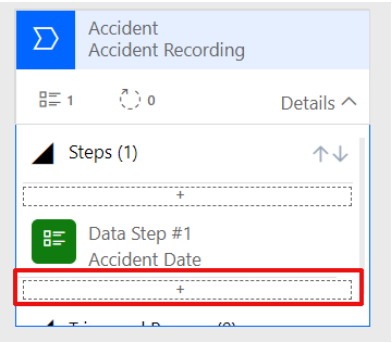
Click Apply:



4.2.6. Click New > Add Data Step:



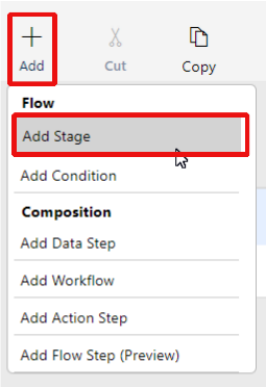
4.2.7. Select the required position of the Data Step:



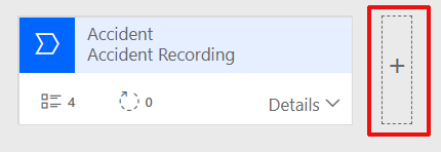
4.2.8. Configure the data step as below:
Step Name: Accident Location
Data Field: Accident Location
Required: Yes
Sequence: 2

4.2.9. Reproduce the same steps for:
Accident Type
Accident Description

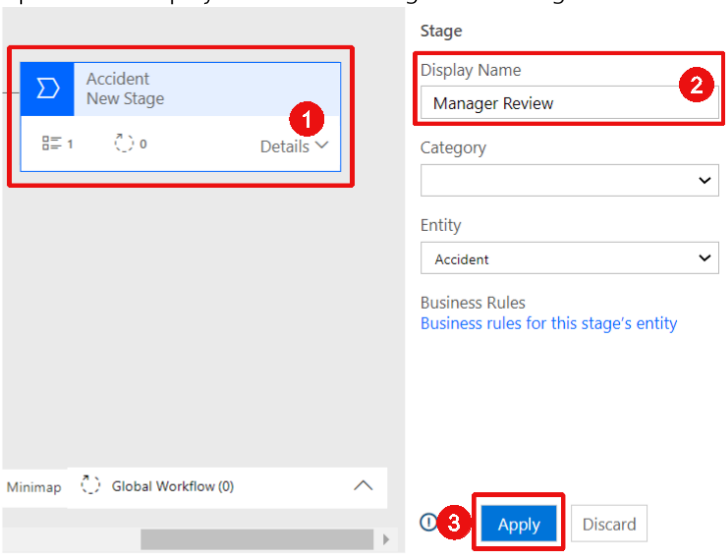
4.2.10. Click Add > Add Stage:



4.2.11. Choose the position of the new Stage:



4.2.12. Update the Display Name of the stage to "Manager Review" and click Apply:



4.2.13. Update the existing Data Step as below:

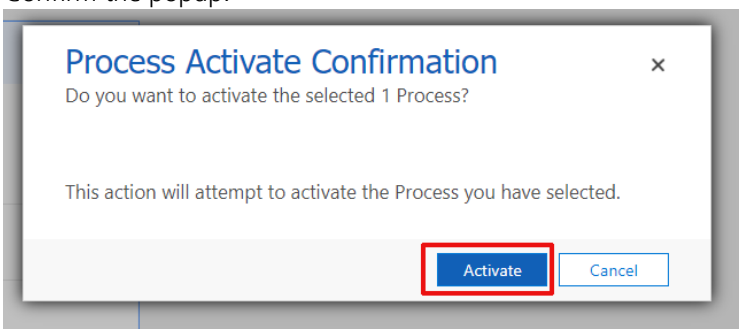
Step Name: Manager Reviewed
Data Field: Manager Reviewed
Required: Yes
Sequence: 1

4.2.14. Create a new Data Step as below:

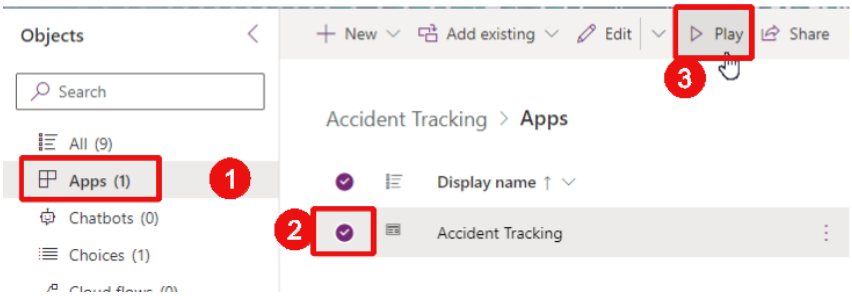
Step Name: Manager Comments
Data Field: Manager Comments
Required: Yes
Sequence: 2

4.2.15. Save and Activate the process:

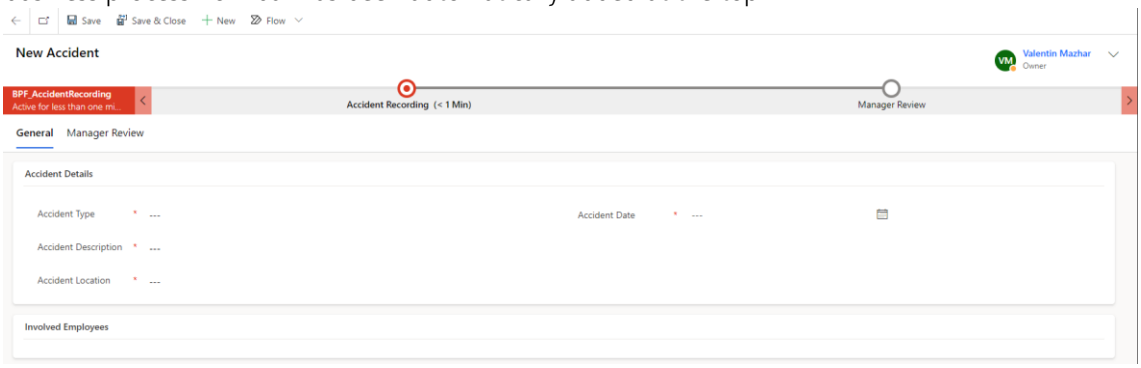
4.2.16. Confirm the popup:



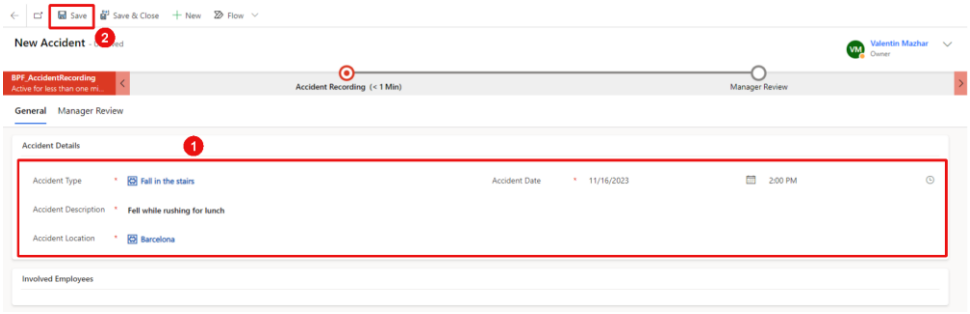
4.2.17. Close the tab and return to your Solution. Select your Model Driven App and click Play:



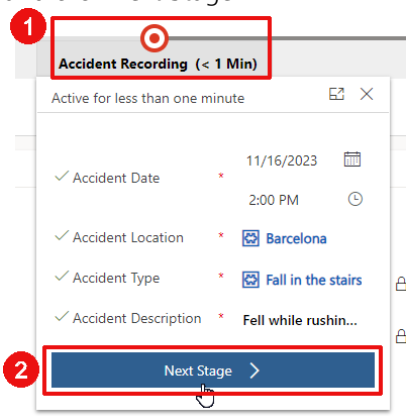
4.2.18. Open the Accidents View and click "New" to create a new Accident. Notice that the business process flow bar has been automatically added at the top:



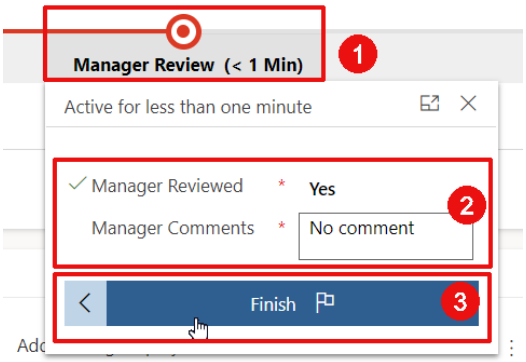
4.2.19. Fill out the required information and click "Save":



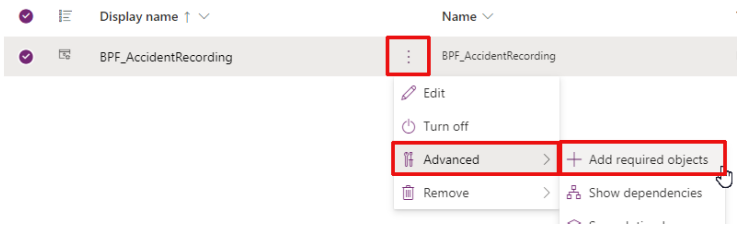
4.2.20. Click on the Accident Recording stage. Verify the information is appropriate completed and click Next Stage:



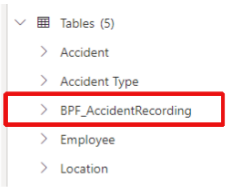
4.2.21. The stage is now moved to the Manager Review. This is when the Manager needs to access the Accident, add a comment and click indicate that the review is completed. Update Manager Reviewed to Yes, add a comment and click "Finish":



4.2.22. Go back to the Accident Tracking Solution.
The business process progress for each Accident is stored in a table automatically created when the business process was saved. That table stores the progress of each Accident in the process. Though the table is necessary for the Model Driven App to work, it is not automatically added to the Solution.
It can be added by selecting the business process flow in the solution, clicking on the 3 dots > advanced > add required objects:



4.2.23. Note that a new table has then been added to the solution:



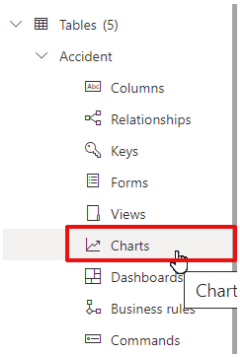
To go further:

- If you want to explore how the data is stored in that table, feel free to configure the Active view and add the related page in the Model Driven App.
- In practice, using the business process table to follow the progress of the Accident recording process would not be recommended. Ideally you would use a Status column in the Accident table and that Status would be updated at each stage. This way you will have all the information needed to follow the process progress in the same table. The Business Process table would only be used in the background to offer the visual process bar for the user.
- The business process we created is very simple. In practice, you could imagine that the Manager review would only be required for high severity accidents. For this, you could add a condition between the two stages that will check the Accident type and only move to the Manager Review stage for severe accident types.

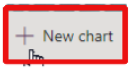
4.3. Create the Charts for the Accident table

It is also possible to embed chats and dashboards within a Model Driven Apps. There is also an integration with Power BI which allows to embed Power BI reports within the Model Driven App, but for these labs we will just create some Charts.

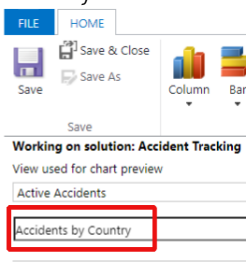
4.3.1. From the Accident Tracking Solution, on the solution tree view panel (on the left), extend Tables > Accident > Click on Charts:



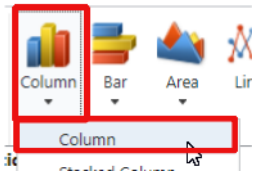
4.3.2. On the top, click "New chart":



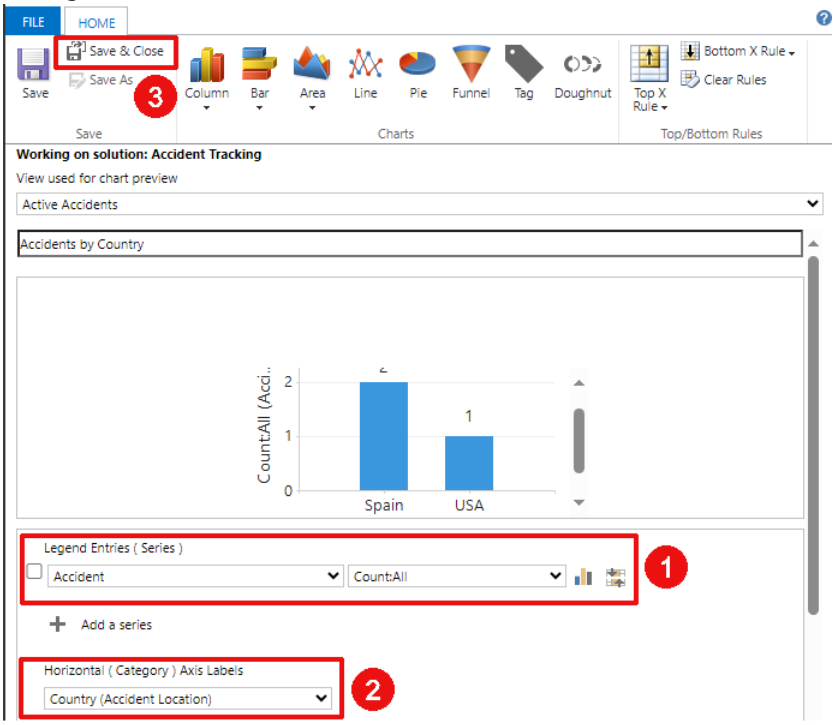
4.3.3. A new browser tab opens for the Charts edition. Add the chart name "Accidents by Country":



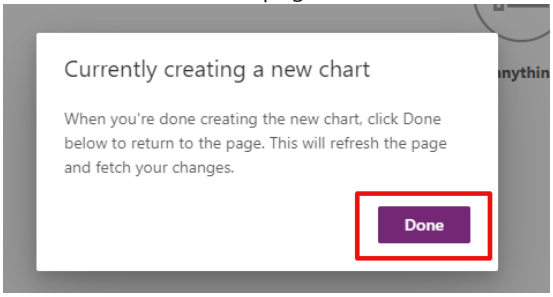
4.3.4. Choose the standard column visual:



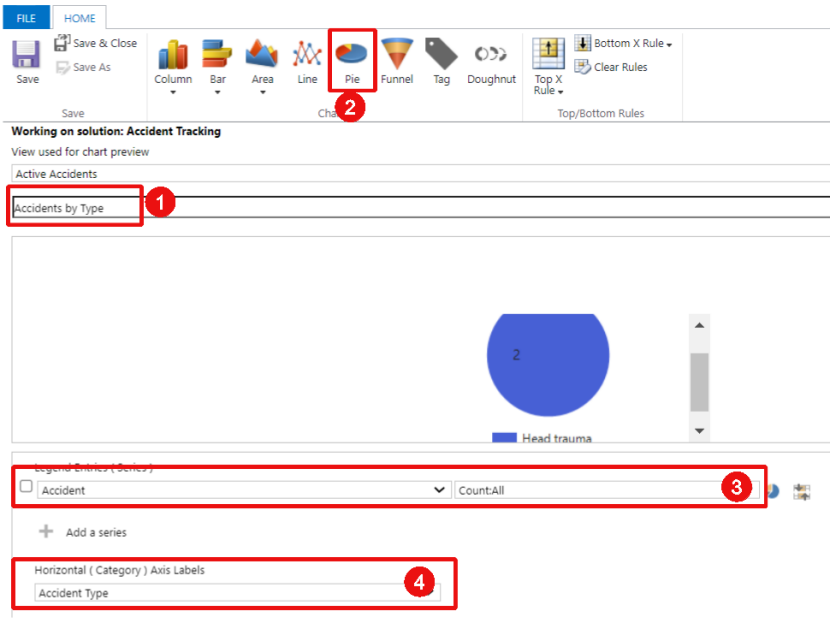
4.3.5. Configure the chart as below and click "Save & Close":



4.3.6. Return to the Solution page and click "Done":

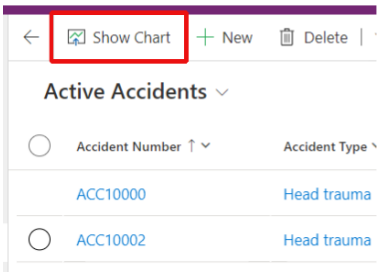


4.3.7. Reproduce the steps to create another Chart to show the Accidents by Type:

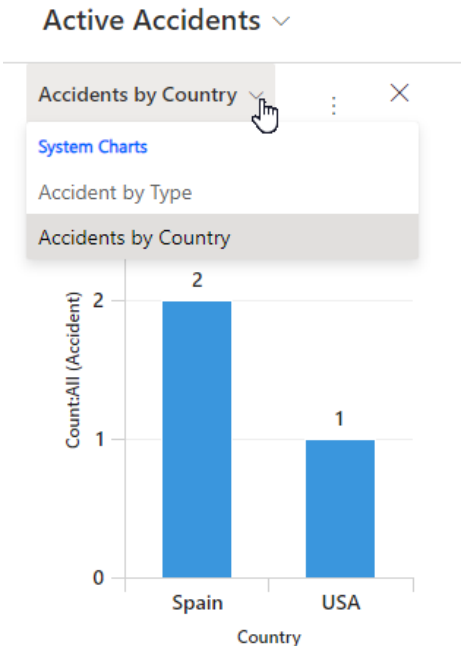


Click Save & Close

4.3.8. Open the Model Driven App and go to the Accidents page. Click on "Show Chart" on the command bar:



4.3.9. Notice the Charts just created:

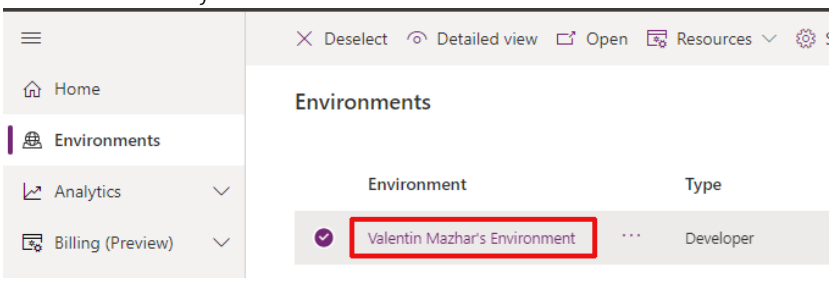


5. Set Security and Auditing

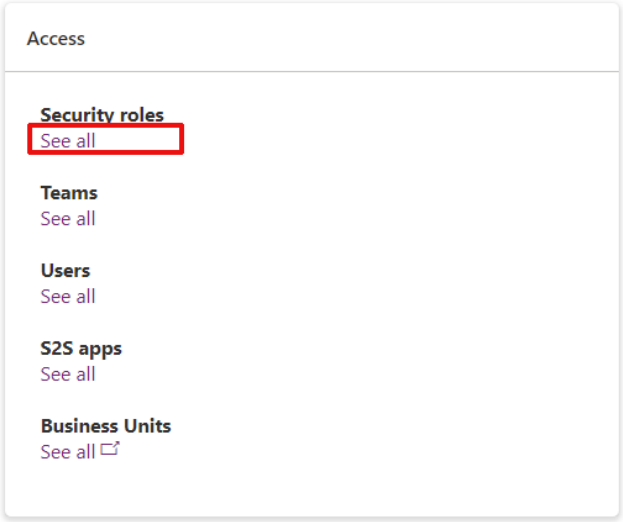
5.1. Create a Security Role

The Dataverse tables that we need are now created. We also have a Model Driven App configured for our Accident Tracking solution. We now need to be able to share the App with other users. In order to do this, we first need to create a Security Role that will grant relevant permissions to the users.

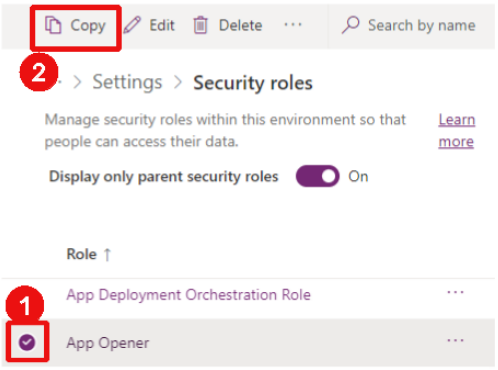
5.1.1. Open the Power Platform Admin Center, identify the environment you have used for these labs and click on your Environment Name:



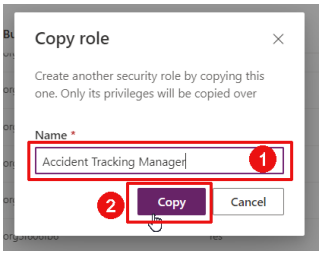
5.1.2. On the right-hand Access panel, open "Security Roles":



5.1.3. Instead of creating a role from scratch, a best practice is to copy an existing role called "App Opener" and customize it. This will ensure that the custom role contains all minimum permissions needed to be be able to run an app. We will then customize it to add the permissions over our accident tracking tables.
Find the role called "App Opener", select it and click "Copy":

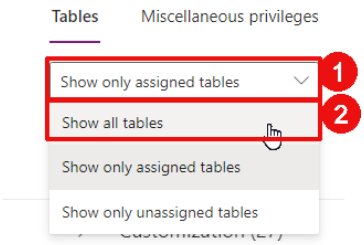


5.1.4. Give it the name "Accident Tracking Manager" and click "Copy":

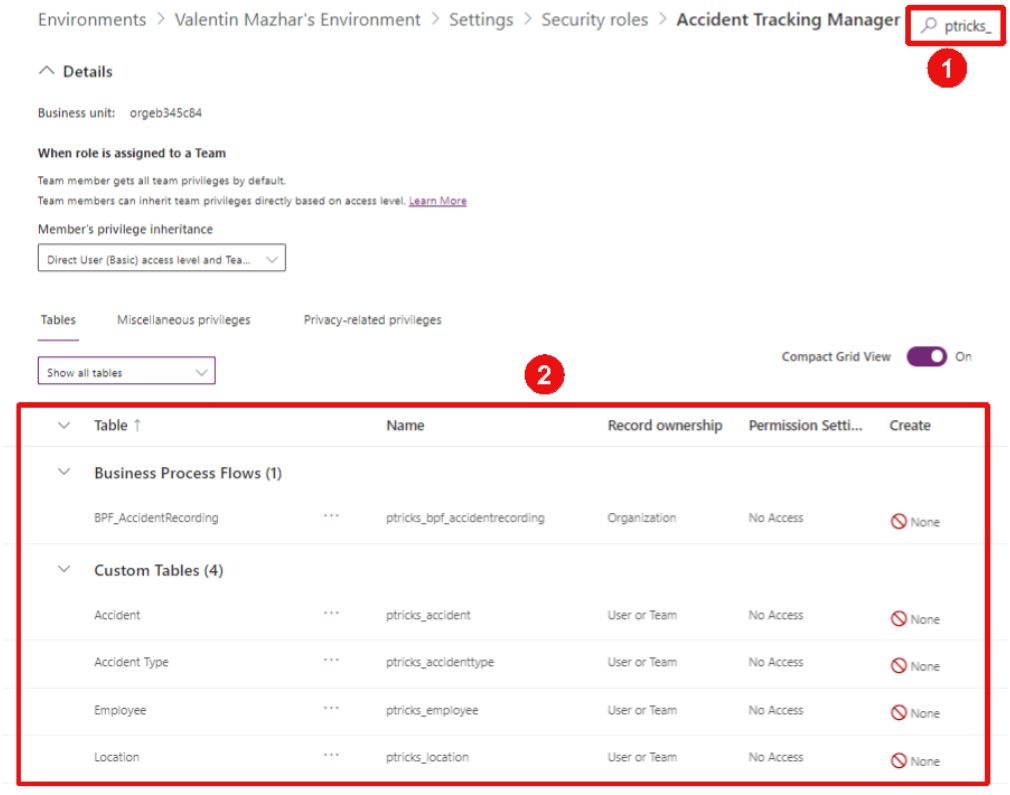


5.1.5. Click on the 3 dots of the created role and click "Edit":

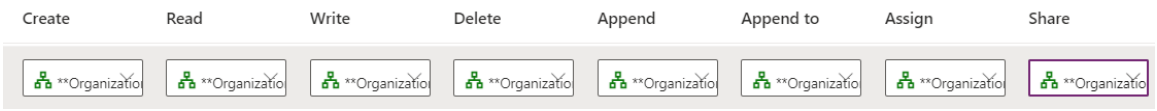
5.1.6. The security role editor has changed recently, you will see the new interface. In some situations you might still encounter the classic interface. On the new interface, you can see the list of all tables and associated permissions for the Accident Tracking Manager role. Select "Show all tables" in the drop down:



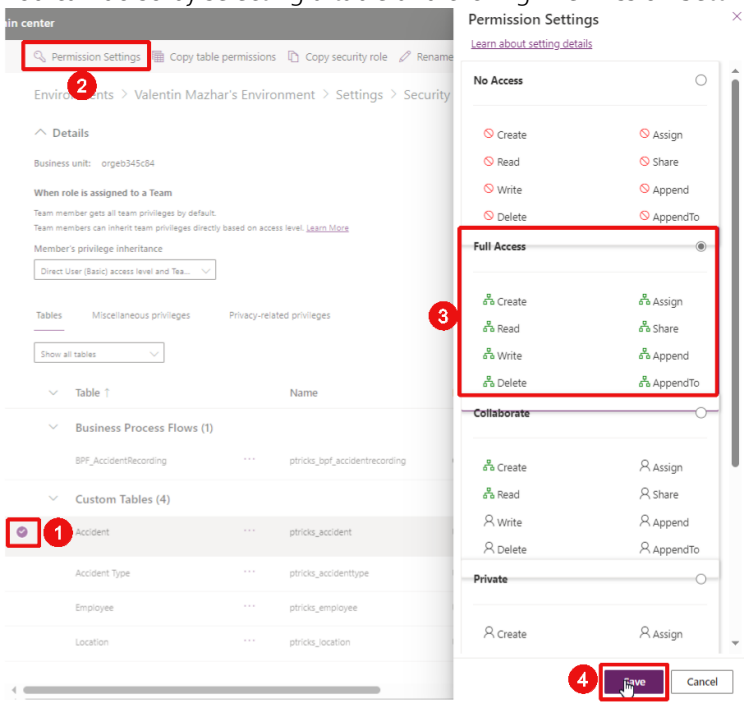
5.1.7. Find all the Accident Tracking related tables. You can search the term "ptricks_" in the top-right-hand search bar to easily identify all the columns related to our solution:



5.1.8. For each table (Accident, Accident Type, Employee, Location, BPF_AccidentReporting), update the permissions as below:

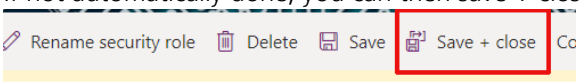


You can do so by selecting a table and clicking "Permission Settings" > "Full Access":

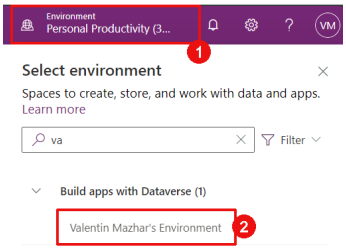


(Notice any interface issues? Microsoft recently updated the design for the security roles, still some work to do there…)

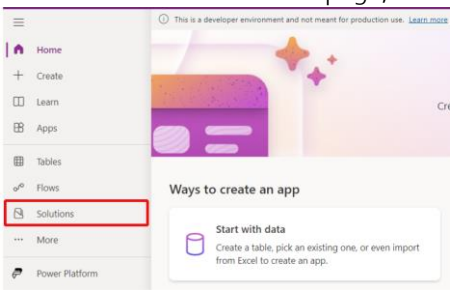
5.1.9. If not automatically done, you can then save + close the role:



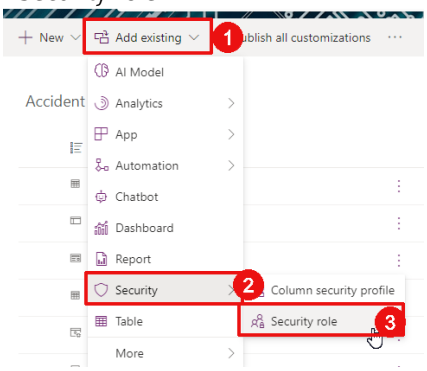
5.1.10. Go back to your Accident Tracking Solution. For this, go to https://make.powerapps.com/home and switch to the environment used for these labs.



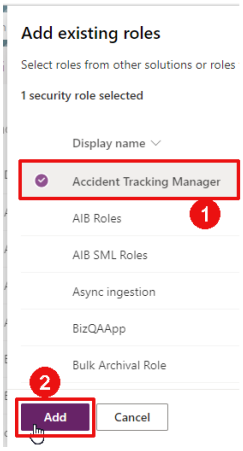
On the left-hand side of the page, click on "Solutions":



5.1.11. We want to add the Security Role to the Solution. This will help us make sure that if we move the Solution to another environment the related security role will also be transported so that we can grant required permissions to users. Click on "Add existing" > "Security" > "Security role":



5.1.12. Select the Accident Tracking Manager role and click "Add":



Notes:

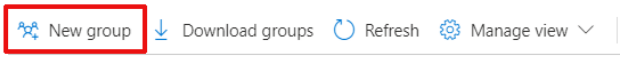
- In order to be able to import the solution into another environment, you will need the System Administrator role on that new environment. This is because only system admins can created / edit / assign security roles. If you do not have such a role in a target environment, an approach could be to have two solutions:
    - 1 solution containing the Tables, Apps and Flows. You will be able to import this solution in an environment with the System Customizer role which is less permissive than the system admin role
    - 1 solution only containing the security role. This solution will be imported by a sytem admin.

  The above approach would allow the admin to not grant an overly permissive role to the project team while still enabling the team to manage their solutions themselves. The security role solution is likely to remain relatively stable which means the admins will not need to update it often.
- In this example we only created one role for the Accident Tracking Manager. This role will grant Organization privilege for all operations to all Accident Tracking tables. It means that if a user has this role they will be able to see, edit, delete all data form these tables. This is therefore a really permissive role. It would be possible to create additional less permissive roles, with only "User" privileges. If a role has a "User" privilege for read and write operations on a table, it means that the user can only see and edit the items that they own. More information about roles and privileges can be found here.

5.2. Assign the Security Role and Share the Model Driven App

We now need to assign the security role and share the Model Driven App with other users. A common practice to proceed is to use a Microsoft Entra ID security group. It is possible to use the same group to share the App and the security role, with only one group to maintain for both.

5.2.1. Open the Microsoft Entra ID and create a new Security Group:

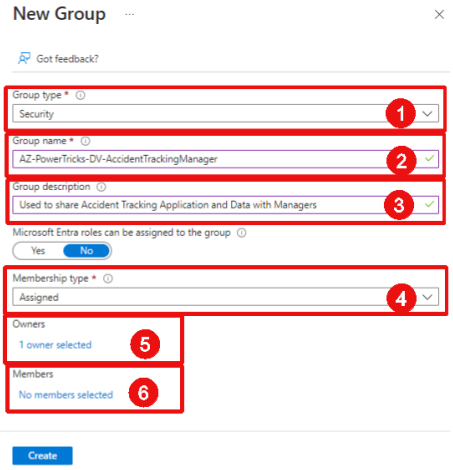


5.2.2. Provide a name for the security group. In these Labs I am using the name below:

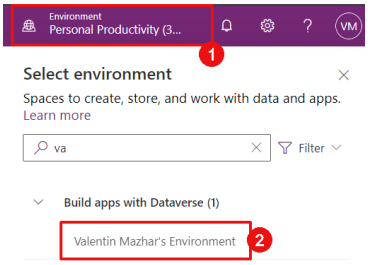
AZ-PowerTricks-DV-AccidentTrackingManager

(Your organization might have some naming conventions to name groups, it is worth asking this information from your Power Platform Admins to make sure you can comply with internal policies. Even if they do not have one, it is still recommended for you to follow one. You can find some suggestions here.)

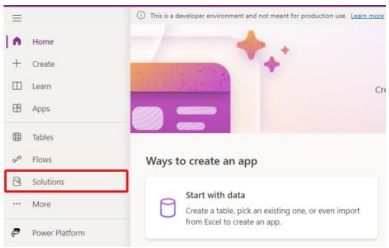
Fill out the fields as below. Since this is only for the sake of these Labs, you do not need to add Owners or Members other than yourself. The Members are Ultimately the users who will have access to the App and data.



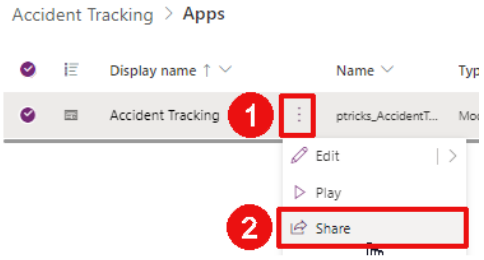
5.2.3. Let's now share the Model Driven App. Go back to your Accident Tracking Solution. For this, go to https://make.powerapps.com/home and switch to the environment used for these labs.



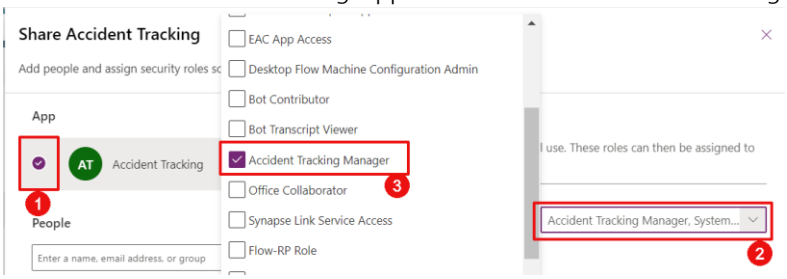
On the left-hand side of the page, click on "Solutions" and open the Accident Tracking Solution:



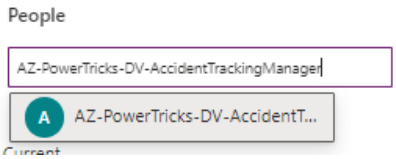
5.2.4. Find the Accident Tracking Model Driven App, click on the 3 dots and click "Share":



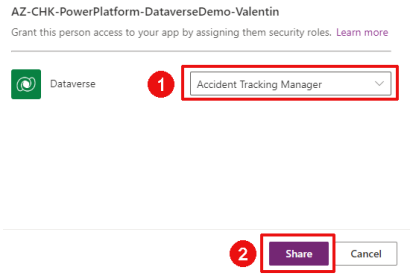
5.2.5. Two steps are required. First, we need to share the App with the appropriate security roles. Click on the Accident Tracking App and select the Accident Tracking Manager role:



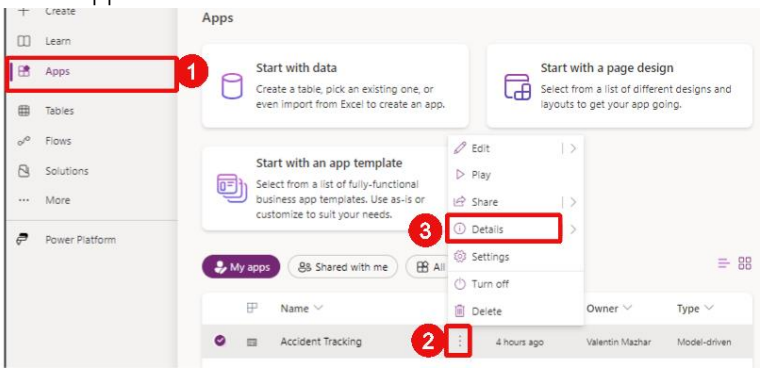
5.2.6. The second step is to share the App with the security group. Seach for the group and select it in the People section:



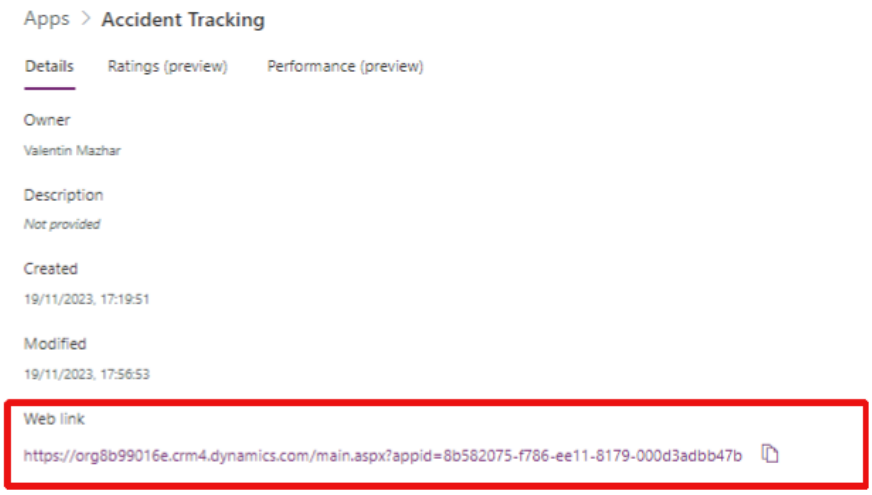
5.2.7. Once the group has been selected, select the Accident Tracking Manager role to assign the role to the users of the group at the same time, then click "Share":



5.2.8. From now on, every time you will add a user as a member of the Entra ID security group, they will automatically be granted the created security role and access to the model driven App. To find the link of the App, open the Apps tab from Power Apps, click on the 3 dots of the App and click "Details":



You will see the link of the App here:

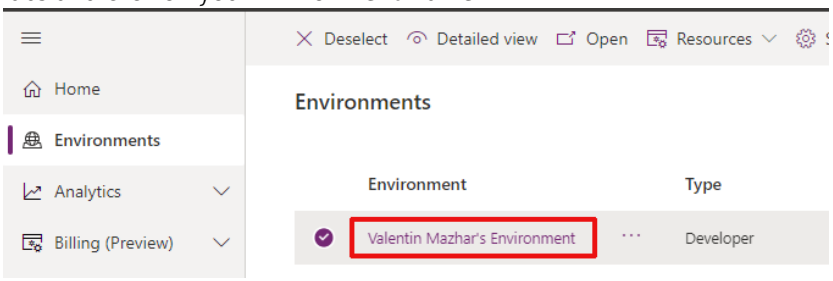


Note: For a user to be able to use a Model Driven App, they also need to be added as a user in the environment. Simply having the App shared and the role assigned to them is not enough. In some environments all users are added automatically without any role but it is not the case of all environments. This means that following the steps of this section will not always be enough to grant someone access to a Model Driven App according to the environment configuration and you might also need to add the user as a user of the environment. This can be done following the process explained here.

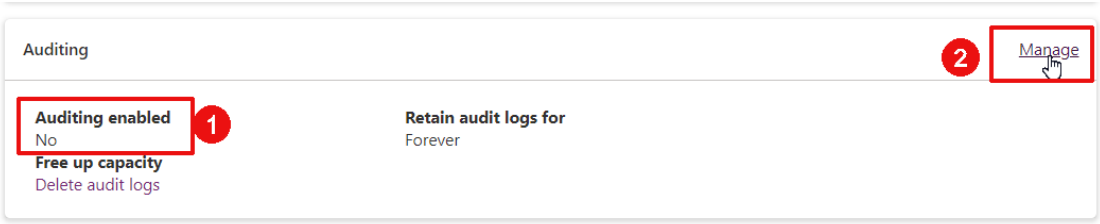
5.3. Turn on Auditing

Let's now look at Auditing. Auditing is a native capability of the Power Platform. Once turned on and configured, it saves any changes made on a table. In order to leverage Auditing, two things need to be done. First, Auditing needs to be turned on for the Environment. Then the relevant tables need to be configured to support auditing (more information here). Let's turn on Auditing on the Accident Tracking tables.

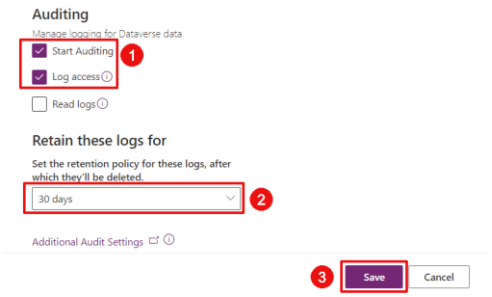
5.3.1. Open the Power Platform Admin Center, identify the environment you have used for these labs and click on your Environment Name:



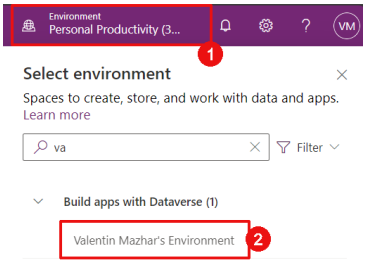
5.3.2. Notice that the Auditing panel shows it is not yet enabled. It is disabled by default. Then click on "Manage":



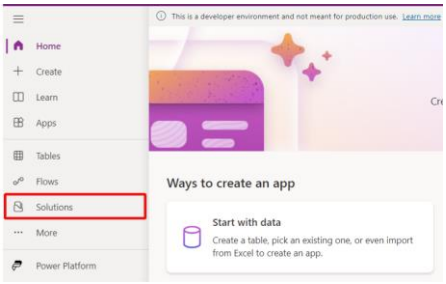
5.3.3. Select "Start Auditing" and "Log Access"and only retain these logs for 30 days. Then click "Save":



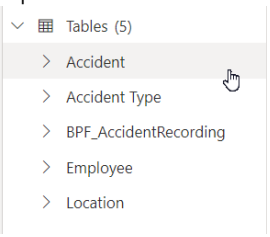
5.3.4. Now let's turn Auditing on for our tables. Go back to your Accident Tracking Solution. For this, go to https://make.powerapps.com/home and switch to the environment used for these labs.



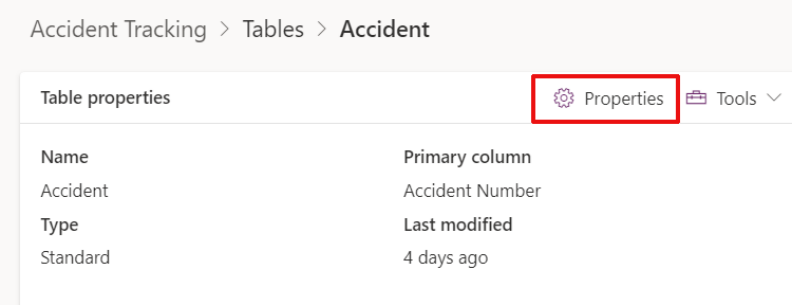
On the left-hand side of the page, click on "Solutions" and open the Accident Tracking Solution:



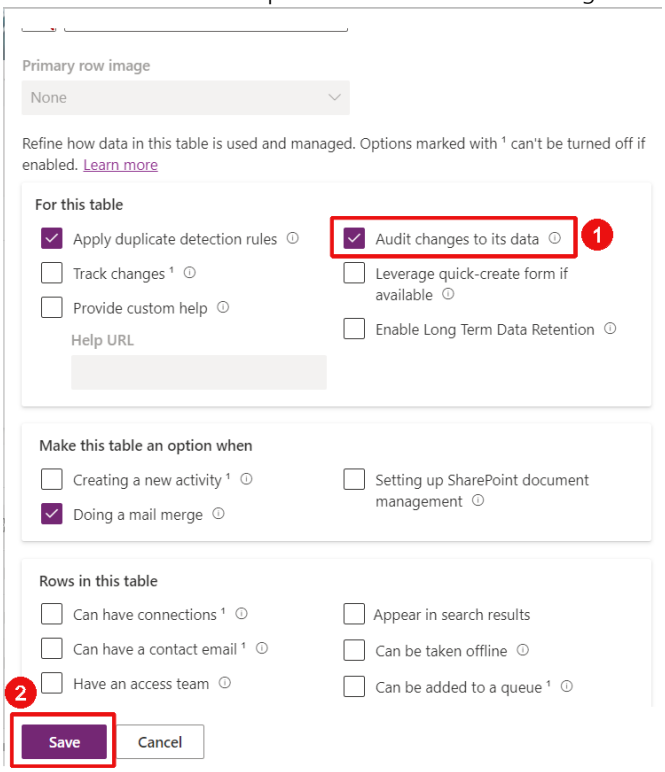
5.3.5. Open the Accident table from the Solution tree view panel on the left:



5.3.6. Open the table properties:



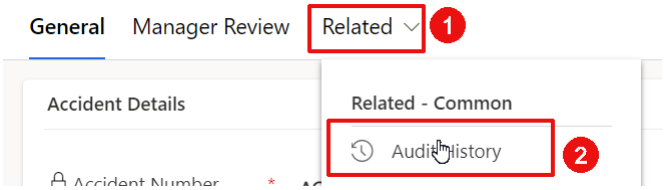
5.3.7. Extend the Advanced options and tick "Audit changes to its data", then click Save:



5.3.8. Reproduce the same above steps for the 3 other tables: Accident Type, Location and Employees

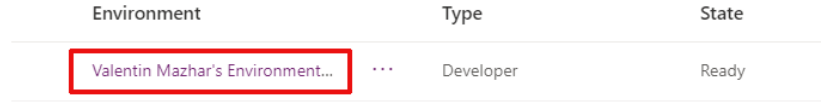
5.3.9. Once this is done, open the Model Driven App and modify an existing Accident record and click Save (for instance by updating the description)

5.3.10. From that same Accident and after having updated and saved it, click on the Related tab to show the Audit History:

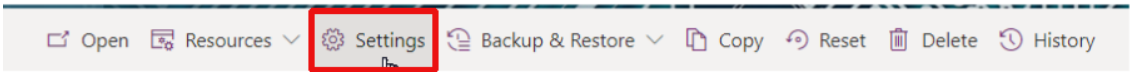


5.3.11. Notice that you will see all the changes made on that Accident. You will even see when the Auditing was enabled for that table:

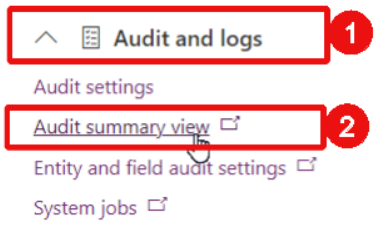
5.3.12. Let's also check the Access Logs for that environment. Open the Power Platform Admin Center, identify your individual environment and click on your Environment Name:



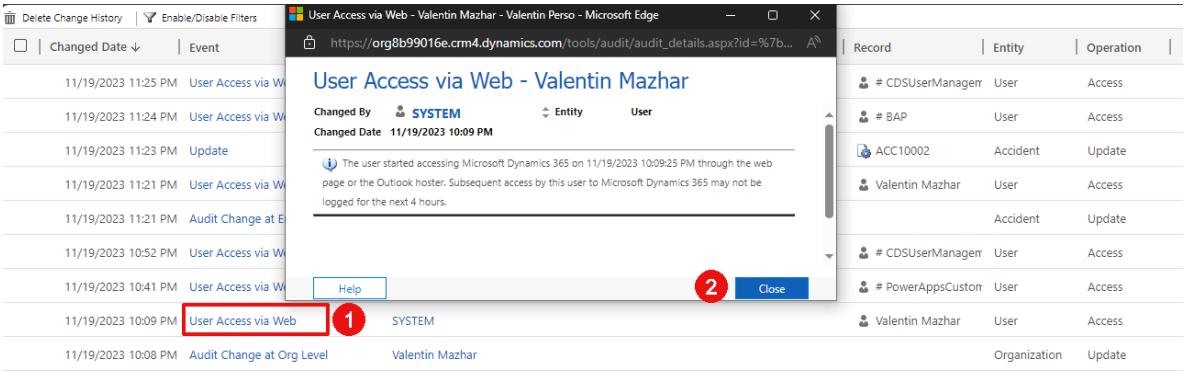
5.3.13. Click on "Settings":



5.3.14. Extend the Audit and Logs section and click on "Audit summary view":



5.3.15. You can then see all the Logs recorded for that environment, including when a user accesses the environment since we turned on the Log access. You can open them for more details:



6. Extend with Power Automate and Power BI

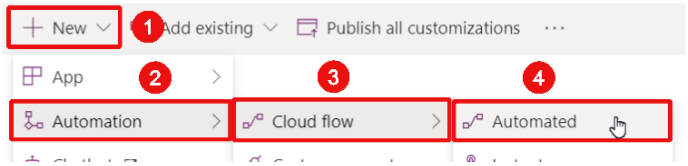
6.1. Create a Flow for notification

Using Dataverse as opposed to SharePoint presents many benefits when working with Power Automate. Essentially the Dataverse connectors offers more options than the SharePoint one.

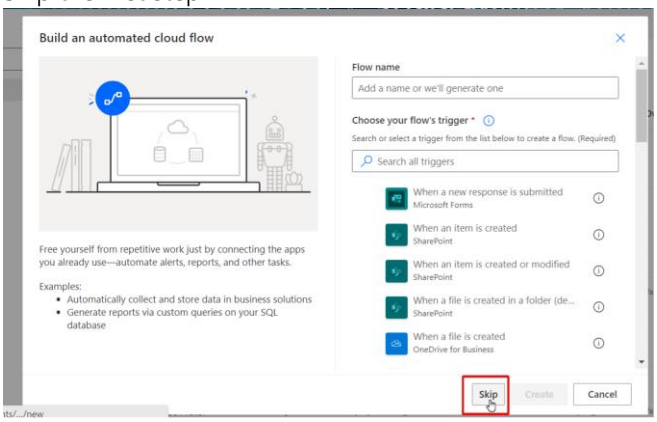
Let's create a simple flow for notification. In the introduction we presented a requirement to notify the creator of the accident including the Head of Safety in CC.

6.1.1. Go to https://make.powerapps.com/home and switch to the environment that you have available for this training. On the left-hand panel, click on "Solutions" and select the Accident Tracking solution

6.1.2. Click New > Automation > Cloud flow > Automated:

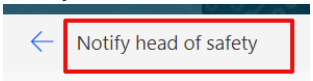


6.1.3. Skip the first step:

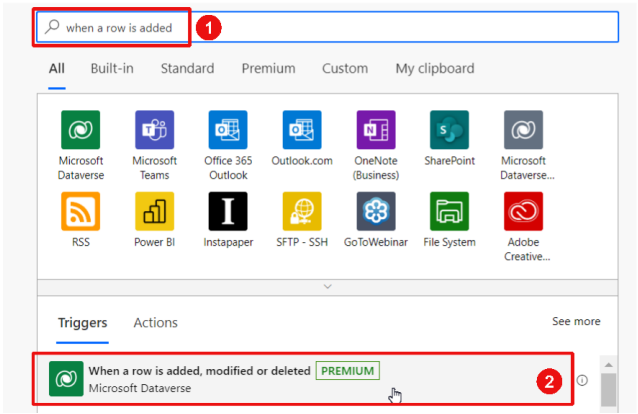


(Creating a Flow from the solution is still using the old interface at the time of writing these labs. It will likely switch to the modern UI leveraging Copilot. For the time being… We will stay in the old UI)

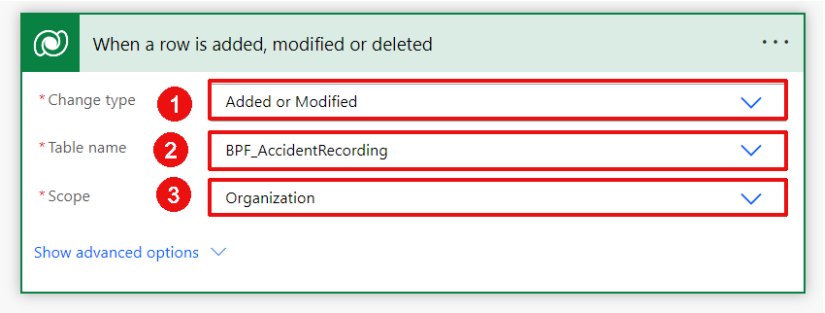
6.1.4. Give it a name on the top-left-hand corner of the screen – for instance "Notify head of Safety":



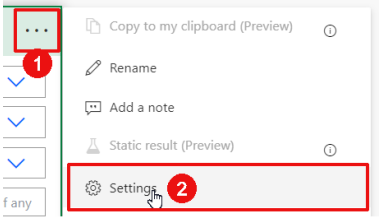
6.1.5. Search for the Dataverse trigger "When a row is added, modified or deleted":



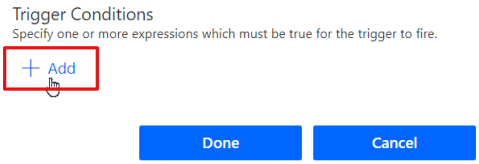
6.1.6. Configure the trigger as below:



6.1.7. If we leave the trigger this way, the Flow will start every time a new process is started or updated. Instead, we only want to send a notification once the Accident has been finalized with the manager review. Therefore, we need to add a "Trigger condition". This condition will need to be met for the Flow to run.
Click on the 3 dots of the trigger > Settings:

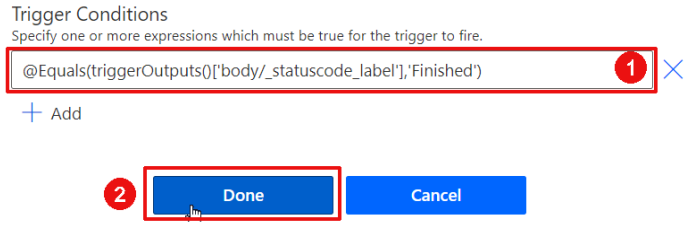


6.1.8. At the bottom of the popup, click "Add" to add a trigger condition:

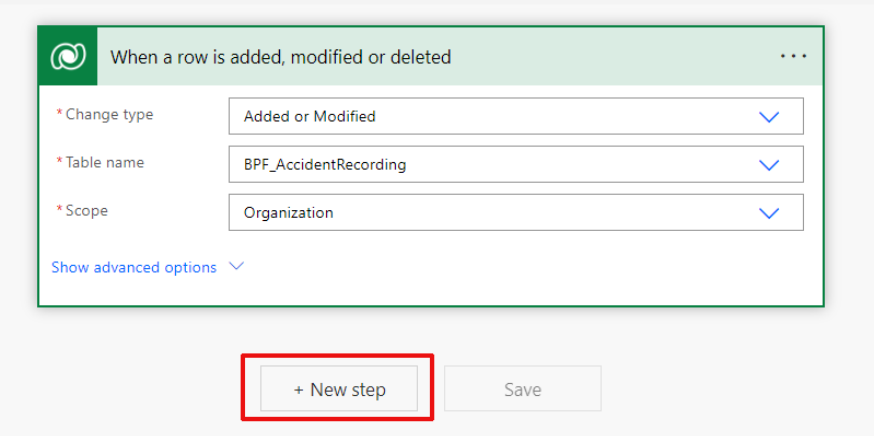


6.1.9. Add the formula below and click Done:

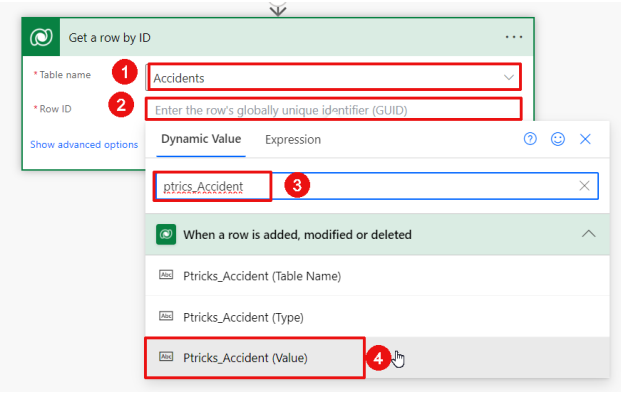
@Equals(triggerOutputs()['body/_statuscode_label'],'Finished')



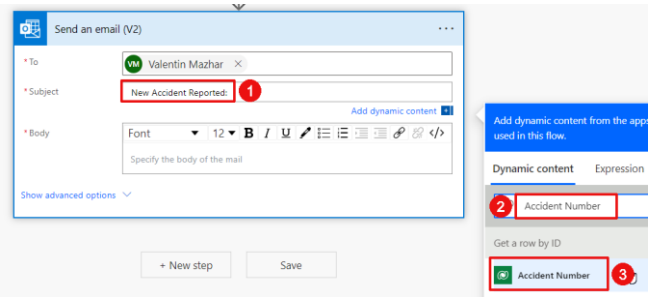
6.1.10. Add a new step in the Flow:



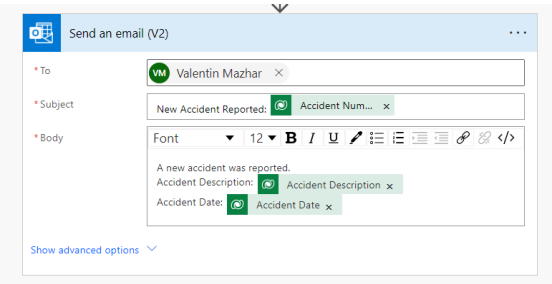
6.1.11. Add the Dataverse action "Get a row by ID". Then configure it as below:
Table name: Accidents
Row ID: select the dynamic content "ptricks_Accident (Value)"



6.1.12. Add another step, this time add the "Send an email (V2)" action from the Office 365 Outlook connector. Configure it as below:
To: select your own account for now as this is just an example. The idea is that you would select the head of Safety instead.
Subject: "New Accident Reported – " and the dynamic content for the Accident Number:

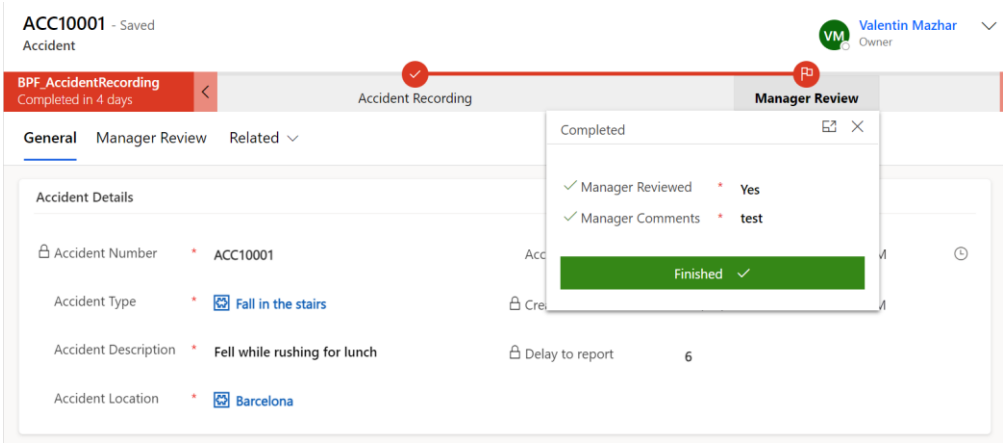


Body: A new accident was reported. Accident Description: [dynamic content for the description]. Accident Date: [dynamic content for the date].
Confirm that the action looks as below:

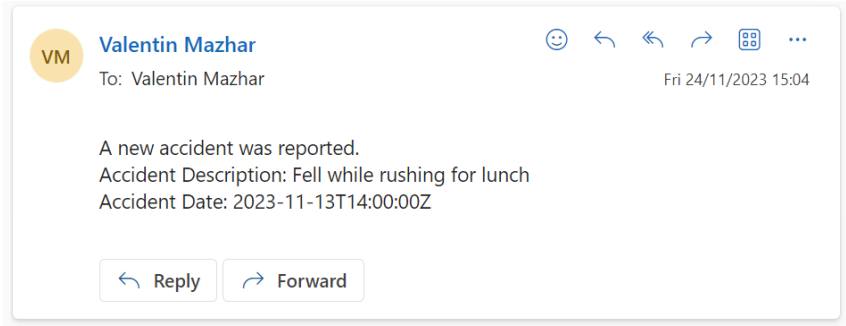


6.1.13. Rename each action by clicking on the 3 dots of each > rename as below, then click "Save":

6.1.14. Open the Model Driven App, create an Accident and complete the Business Process Flow until the end:



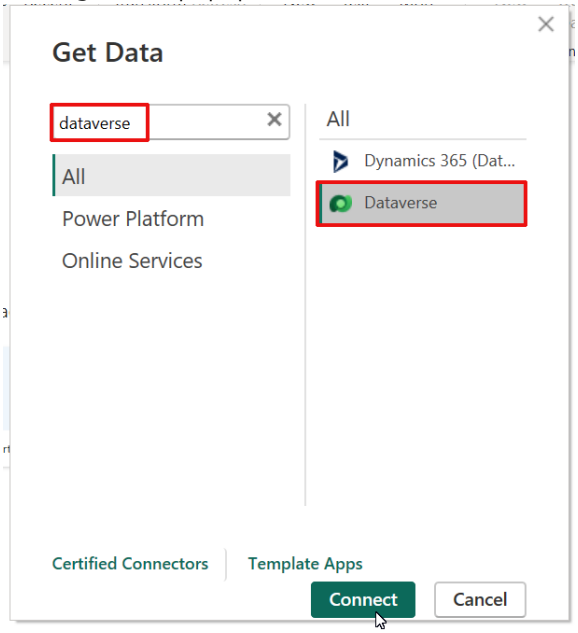
6.1.15. Confirm that you have received an automated email:



6.2. Connect to Dataverse with Power BI

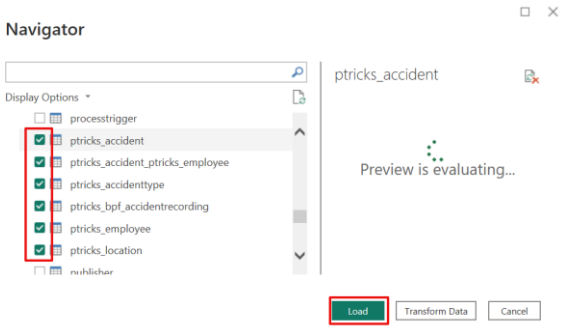
Since this is a Dataverse training we will remain high level on the explanations to connect with Power BI and will consider that you already know how to create a Power BI report.

6.2.1. Open Power BI Desktop to create a new report
6.2.2. In the get data popup, search for Dataverse and click Connect:

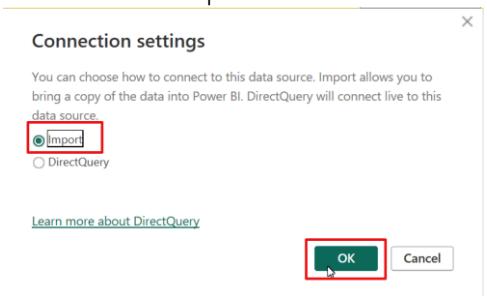


6.2.3. You will now see all the Power Platform environments that you have access to. Expend the environment you have used for the training and select the tables below:

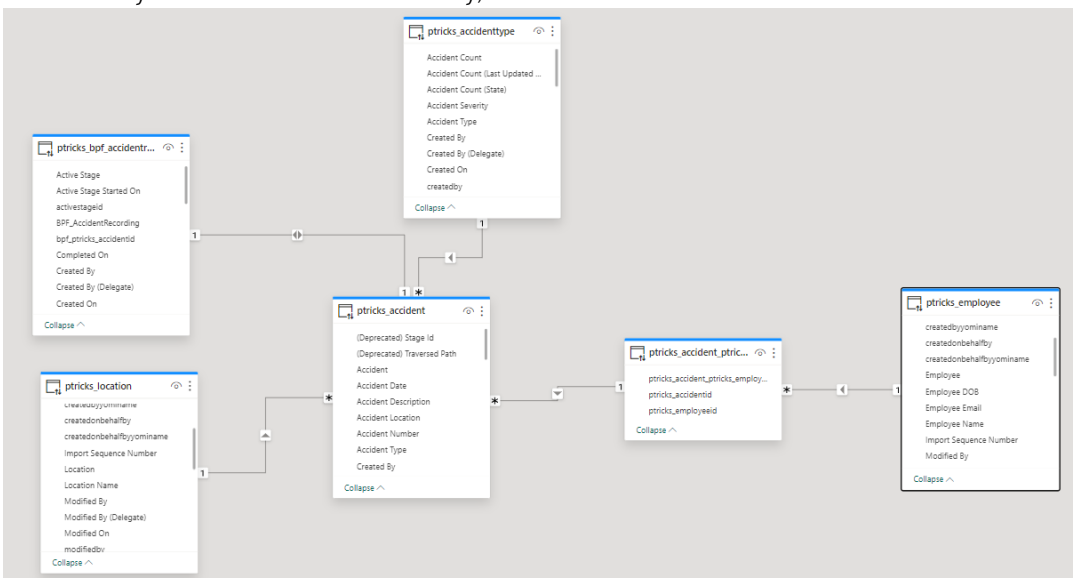
ptricks_accident
ptricks_accident_ptricks_employee (notice that this is the hidden table created when we created the Many-to-Many relationship between Employee and Accident
ptricks_accidenttype
ptricks_bpf_accidentrecording
ptricks_employee
ptricks_location



6.2.4. You can choose the connection mode between Import or Directy query. For this example, we will choose Import:



6.2.5. The data is now loaded into the report. If you go to the Data Model tab, you will notice that the relationships have automatically been defined (unless this setting is turned off in which case you can create them manually):



6.2.6. You can then proceed with creating the report as you would usually do for any other data source.

Conclusion

Congratulations if you have gone through the entire Labs!

If you have, then you have gone through all the foundational Dataverse capabilities. There is a lot more to Dataverse than what has been covered here though, and I sincerely hope these labs will encourage you to continue exploring this great platform.

I would love to hear your feedback about this training or anything else related to PowerTricks. Please reach out via the PowerTricks Contact section, creating an issue in GitHub or directly via LinkedIn!