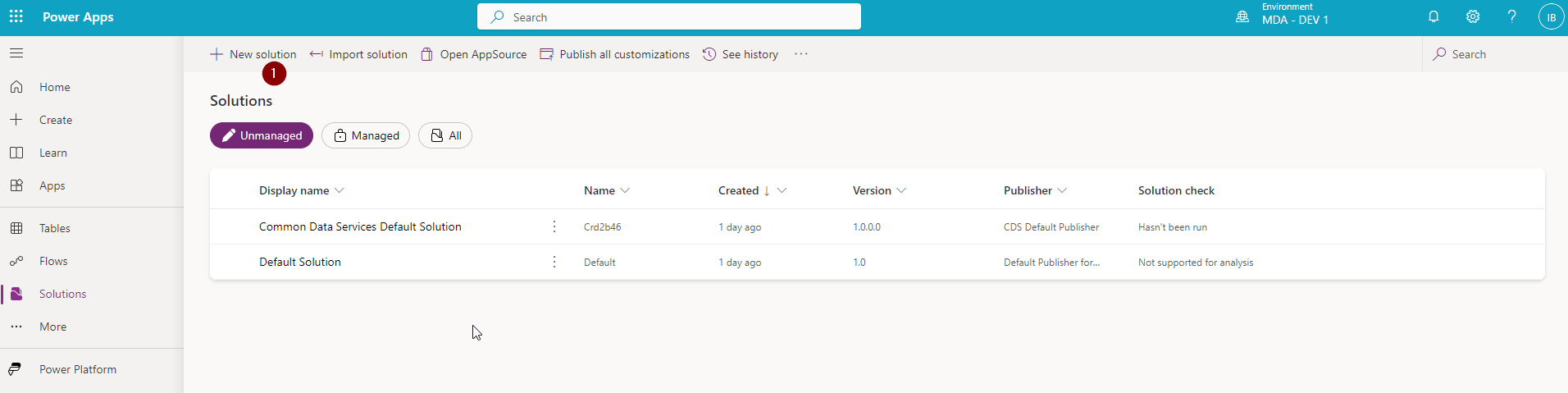
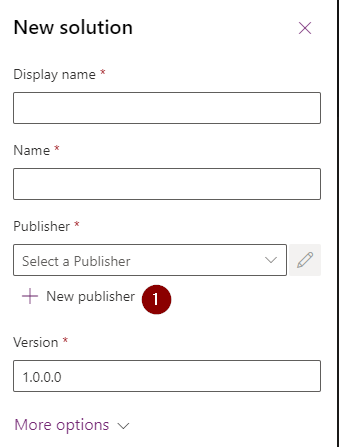
Once we’re done with the Environment setup, we can enter the path of adventures, fun and glory, and start creating our Events manager solution.

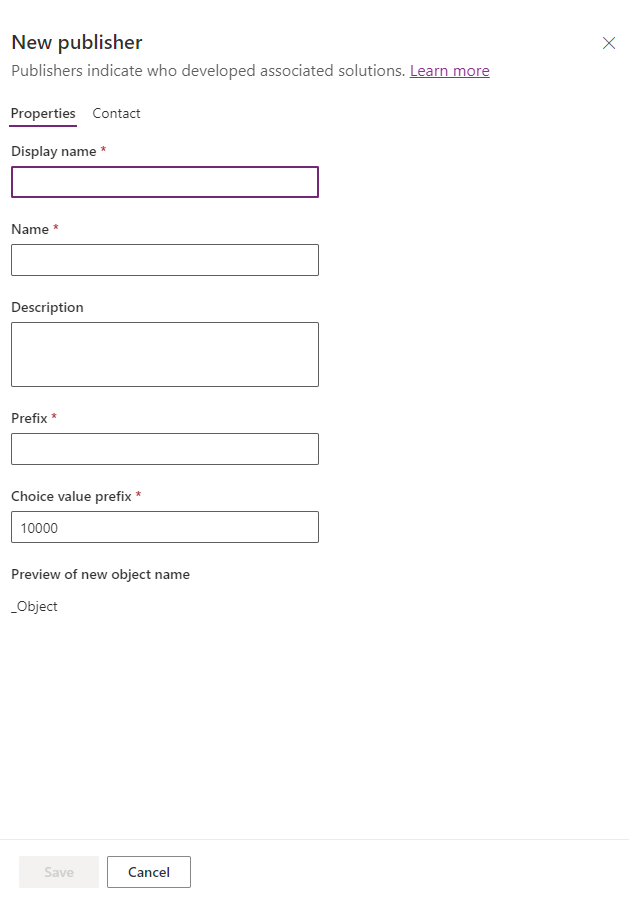
**CREATING NEW SOLUTION AND NEW PUBLISHER**



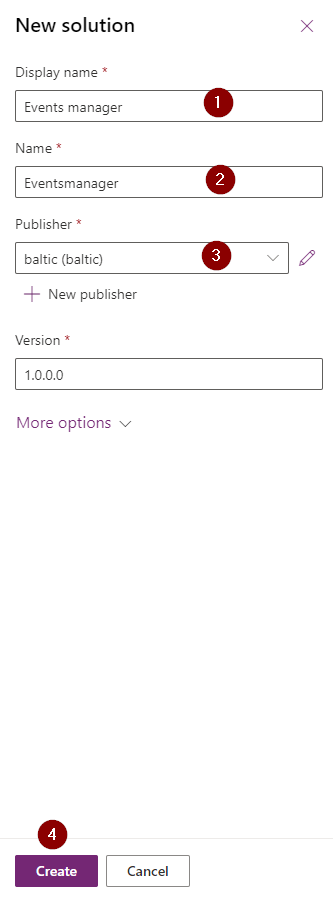
1. Click on the **New solution** button.
2. A new window will open.



If you haven’t created Publisher already, click on the New publisher button.



Enter the Publisher data.



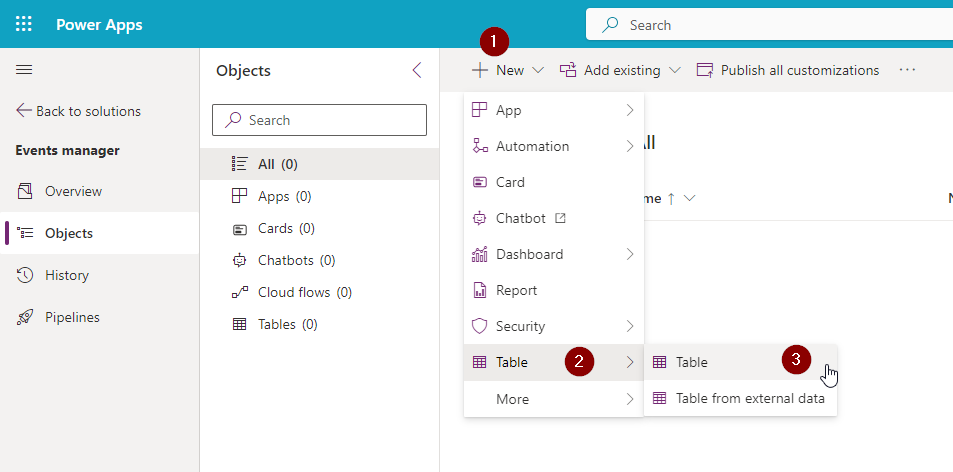
Once you’re done with creating the Publisher, go back to solution creation.

1. Enter Name.
2. Enter Display name (it derived automatically from solution’s name, but you can change it here).
3. Select the Publisher.
4. Click on the Create button.

And there you go – you just created your first solution!

Unfortunately (or fortunately if you like to make your hands dirty) that’s not enough ☹

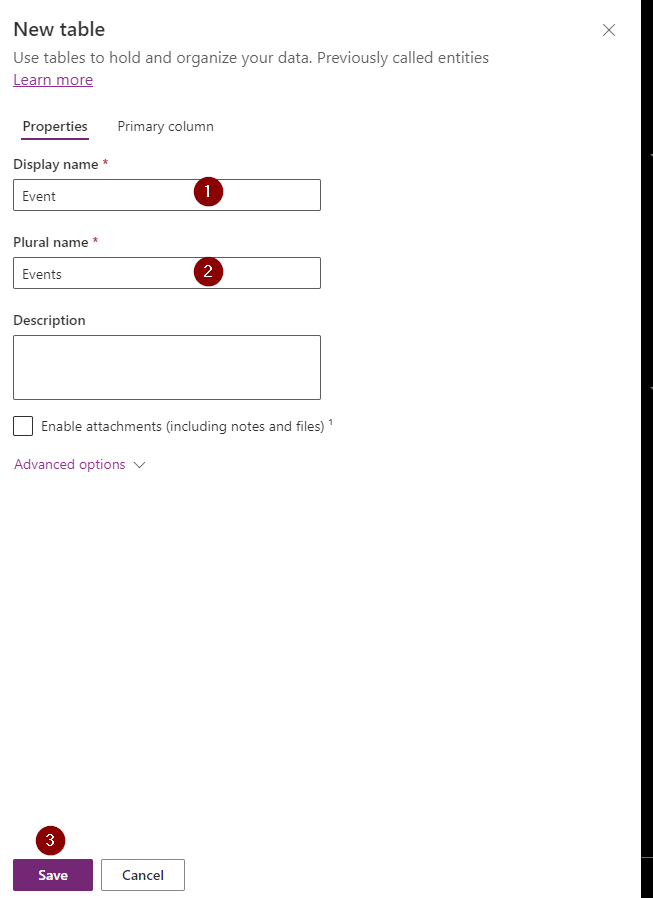
**CREATING NEW TABLE**

****

Open solution you just created. Click on:

1. New.
2. Table.
3. Table.

The pop-up will arise, making your heart beating faster.



Enter:

1. Display name.
2. Plural name (unless it doesn’t make any sense, better leave it as it is – derived from Display name).
3. Click on the Save button.

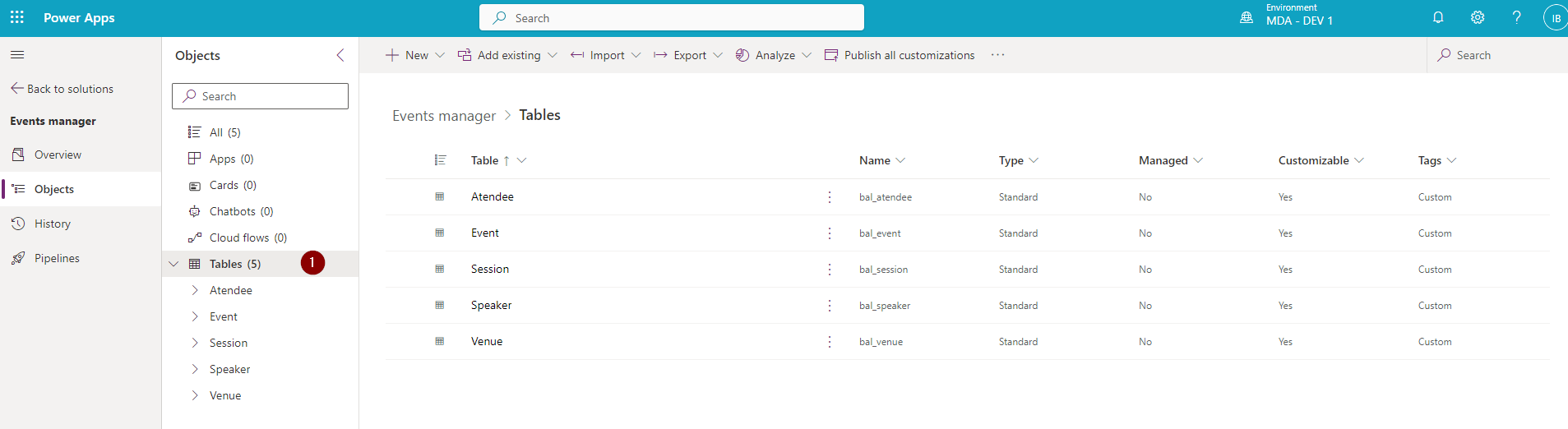
That’s the minimum effort. If you consider yourself a maximizer, you can also go to Primary column tab and change primary column name (default name of primary column name is… Name! What a strange world we live in!).

**CREATING COLUMNS**

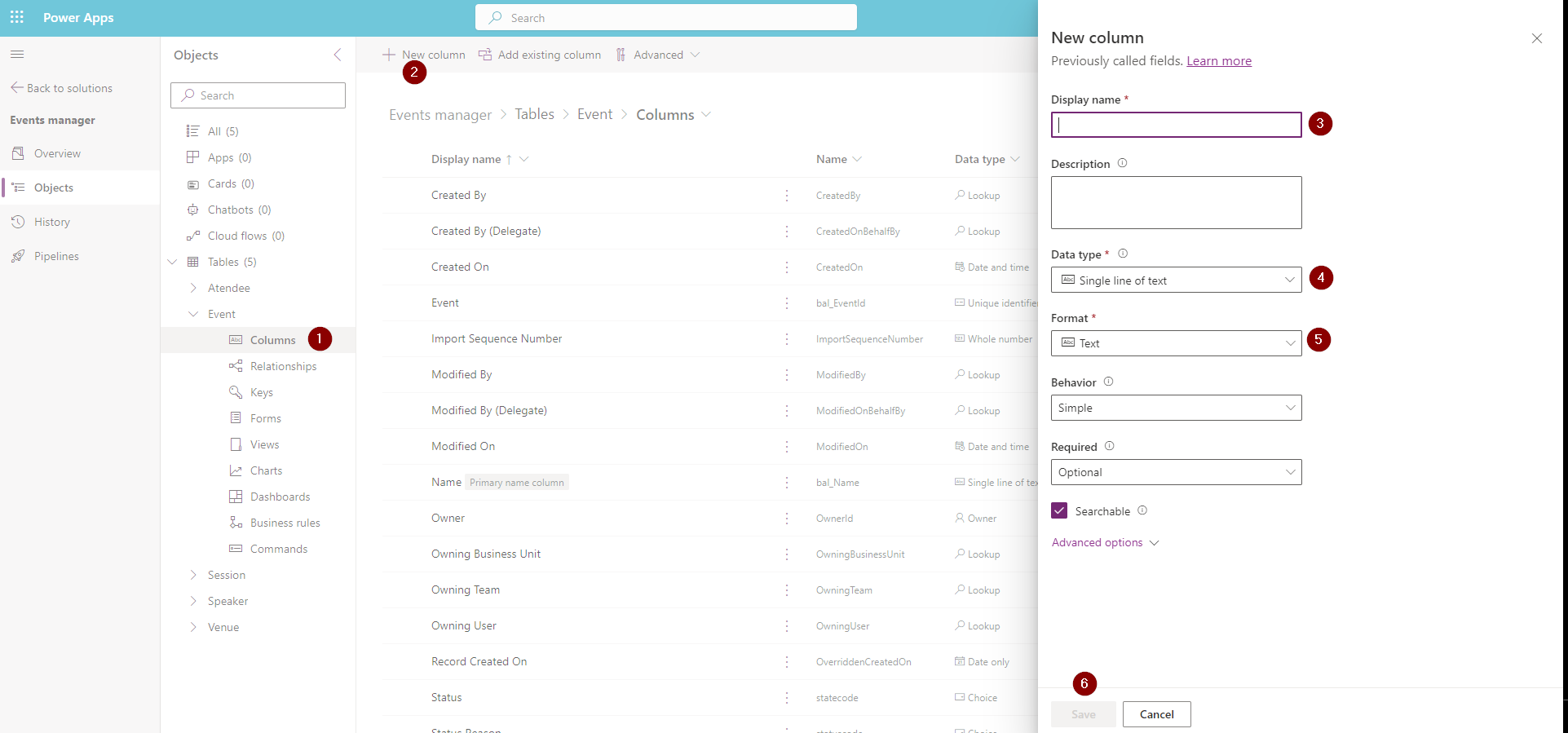
If you studied history of art briefly, you probably know that in ancient Greece there were three types of columns: Doric, Ionic and Corinthian.

But it’s the Dataverse, and not the ancient Greece we live in! So, there’s much more. You can find their description here: [Types of columns](https://learn.microsoft.com/en-us/power-apps/maker/data-platform/types-of-fields).

Let’s create some right now.



1. Go to Tables (you see that I’ve already created some additional ones) and select Event table.



1. Click on Columns.
2. Select New column. A side pane will open on the right.
3. Enter column’s Display name.
4. Select Data type from the dropdown (if you followed the link above you should be familiar with column types. If not, why am I even writing this manual?).
5. Select Format.
6. Click on the Save button.

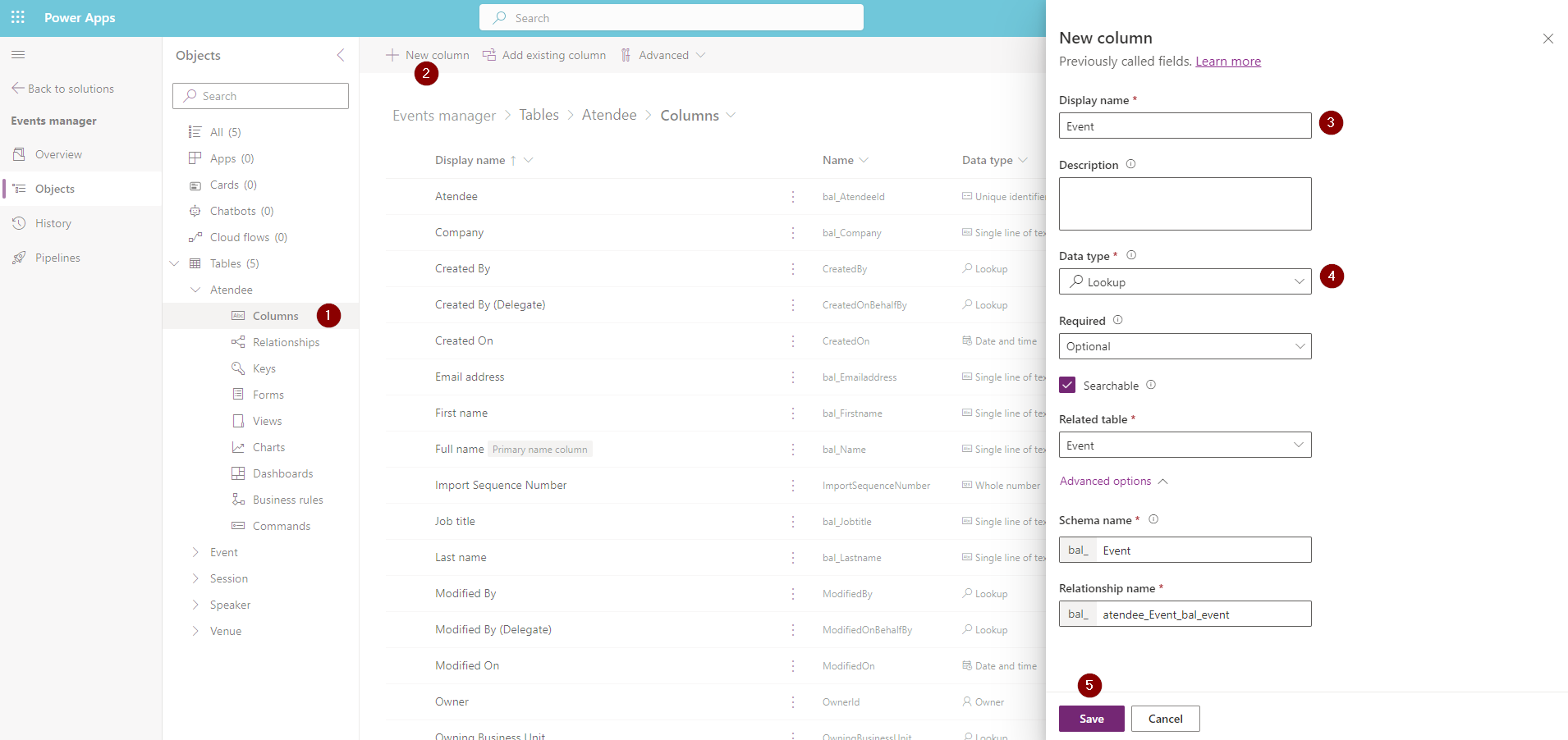
You can also define some Advanced options, Behavior and whether the column is Required. You can, but you don’t have to.

**CREATE LOOKUP COLUMN**

Is lookup column a special column? Aren’t we all special?

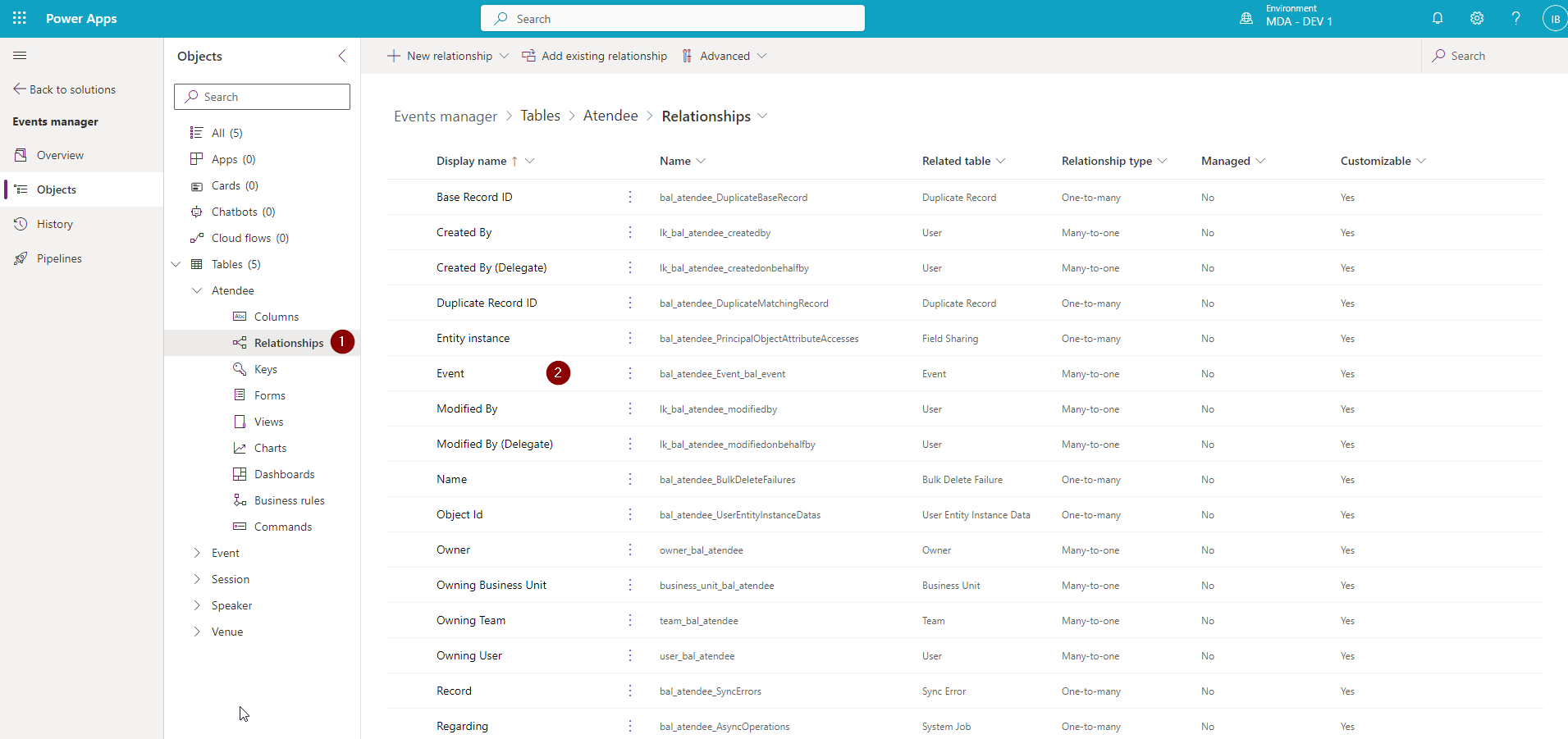
A Lookup column is a column which allows us to create a relationship with another table. The relationship with that table is n:1 relationship. Below is an example. In the Attendee table I created an Event lookup meaning that each Attendee is assigned to a specific Event.

How to create a lookup column?



1. Select the table and go to Columns.
2. Click on New column.
3. Select column Display name.
4. Set Data type to Lookup and select related table.
5. Click on the Save button.

Now you can go to the Relationships section of that table.



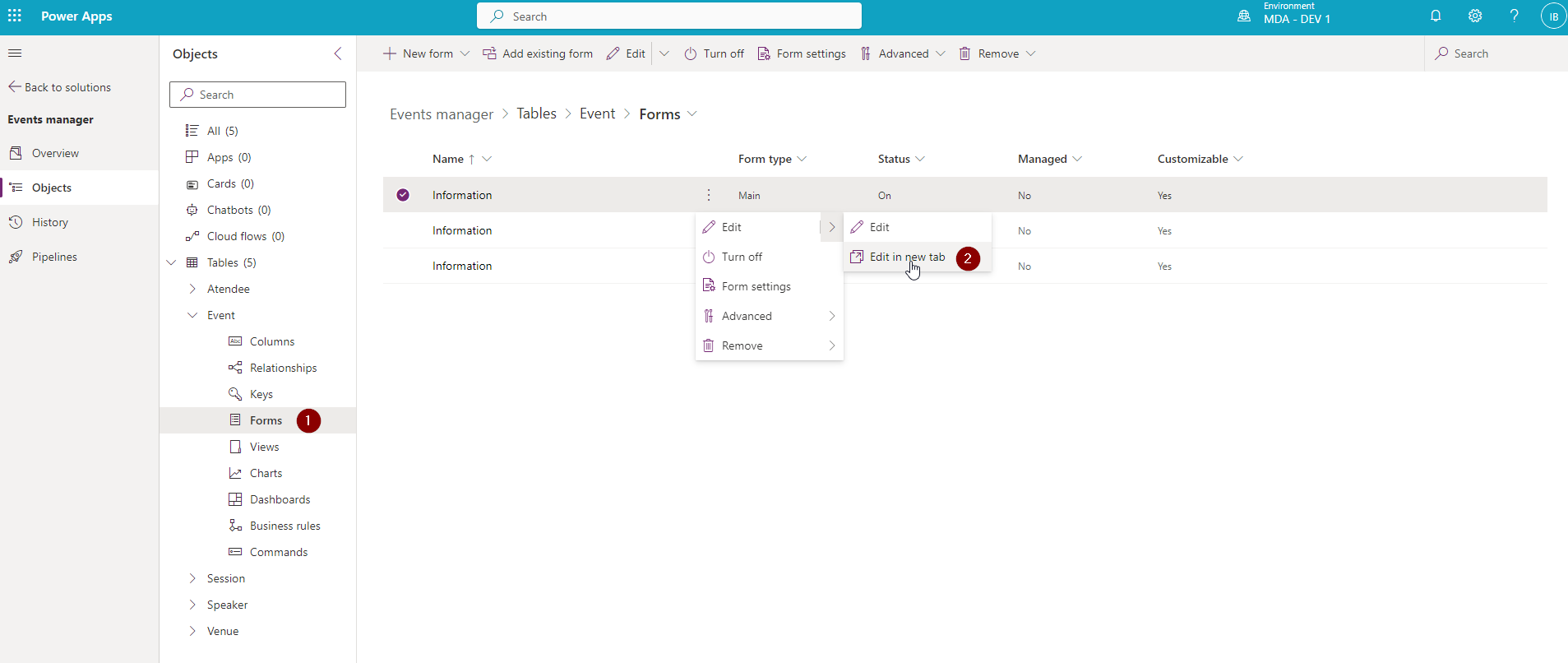
1. Click on Relationships.
2. As you can see the relationship to the Event table was automatically created.

Maybe life isn’t that tough after all?

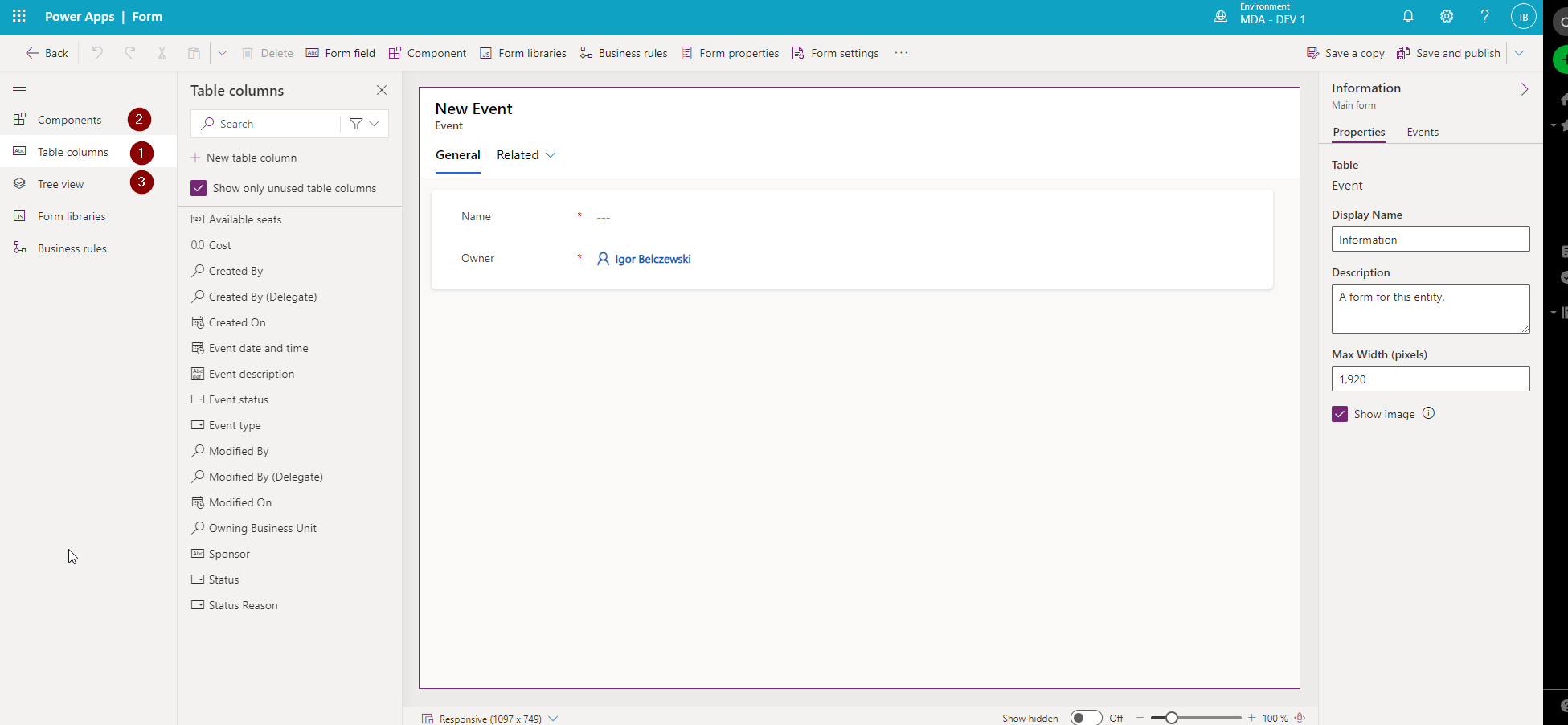
**CREATE AND EDIT FORMS**

Let’s create some forms. Wait, what? They’re already there?!

Ok, change of plans. We’re gonna edit the existing form on the Event table.



1. Go to Forms. As you can see the guys behind the naming weren’t very creative, and we ended up with three forms called Information. Fortunately, they have different types.
2. Click on the Information form of the type of Main and click Edit in a new tab (cause as we all know: the more tabs, the better).

Your eyes will be pleased to see new tab with the form editor.

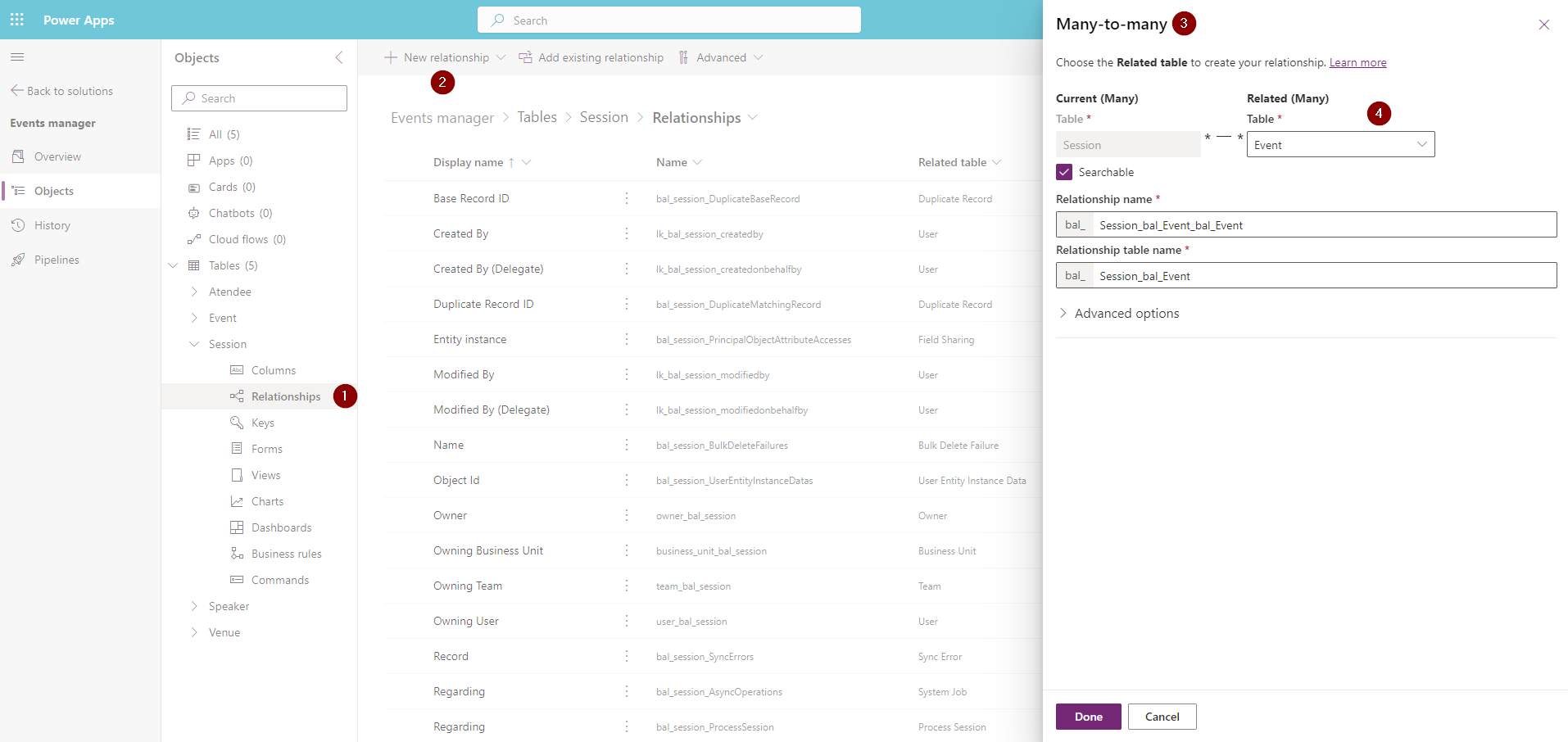
Here you can:

1. Add Table columns.
2. Add Components.
3. See the Tree view.

As you can see, the possibilities are almost endless. Play around and make your form great again. Just don’t call me crying if you break something.

**CREATE MANY TO MANY RELATIONSHIP**

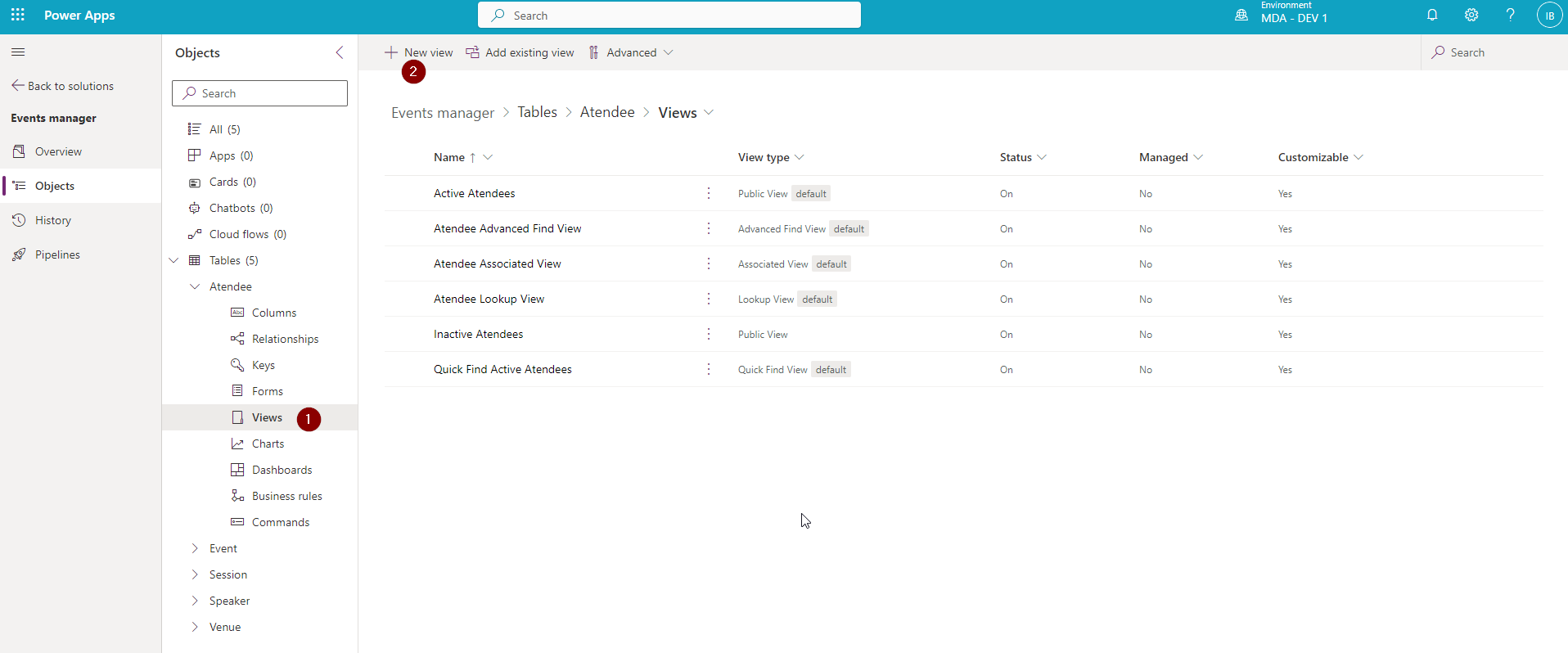
If you find it hard to sleep at night don’t dig into many too many relationships. Cause then you won’t sleep for sure. But if you sleep like a baby, just take a look at what it is all about: [Create many-to-many relationships.](https://learn.microsoft.com/en-us/power-apps/maker/data-platform/create-edit-nn-relationships) And then create one.



1. Go to Relationships.
2. Click New relationship and select Many-to-many.
3. Define related table and click on Done button.
4. Sleep.

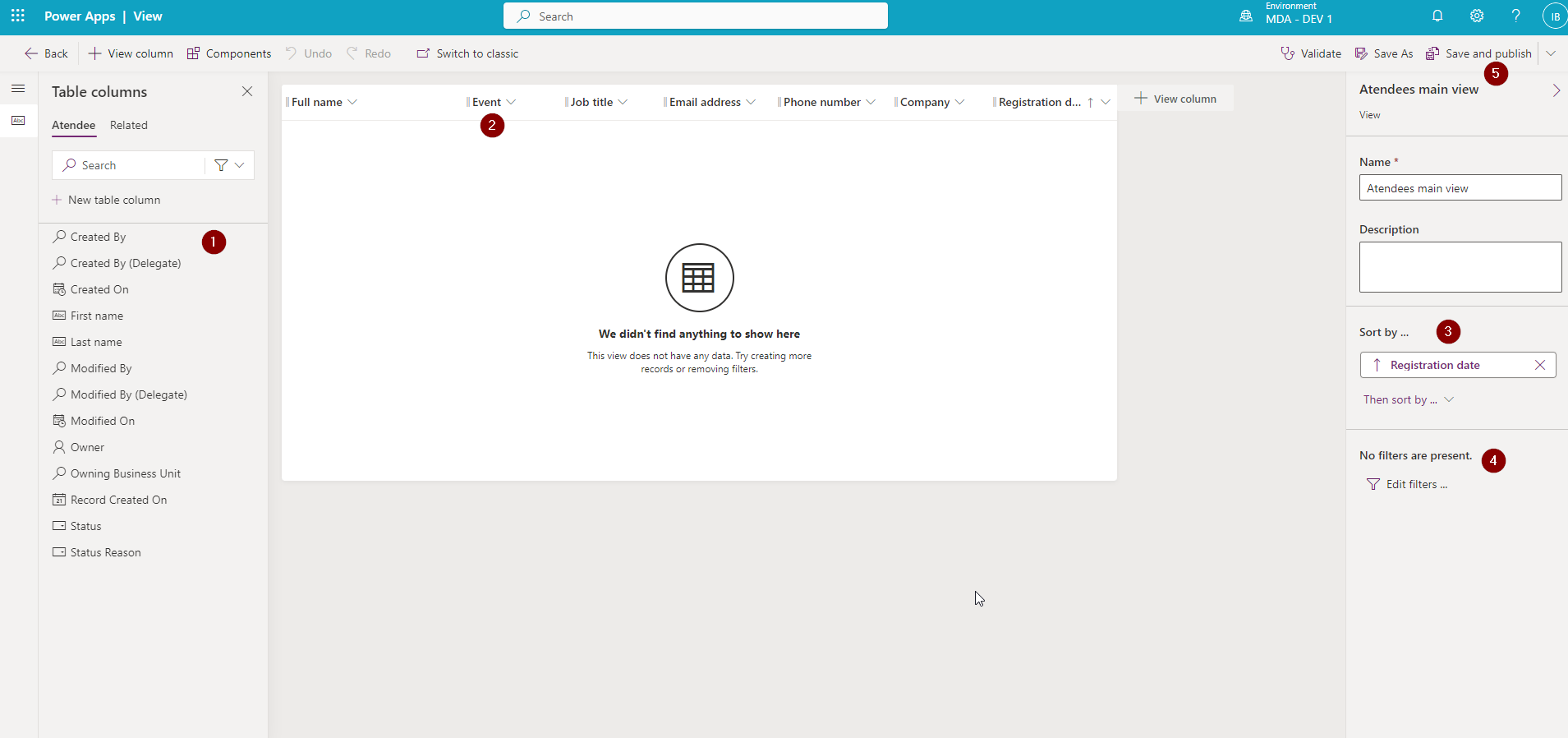
**CREATE AND EDIT VIEWS**

One's point of view depends on the point where one sits. And there you are, sitting in from of your computer. Let’s create some views to show them all!



1. Go to Views on the Attendee table.
2. Click on New view.

View editor will show.

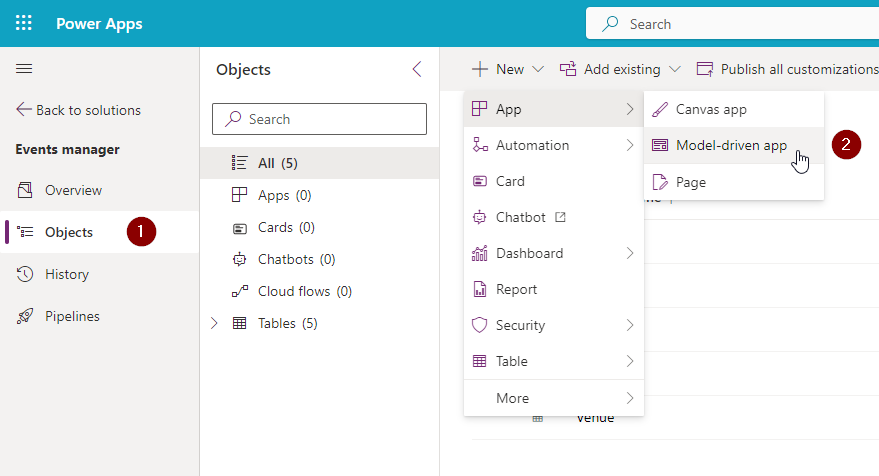


1. Add the columns you need.
2. Order them.
3. Sort them if you want.
4. Add a filter if you want.
5. Save and publish in order to save and publish.
6. Sleep again.

See, it’s not that hard if you believe in yourself.

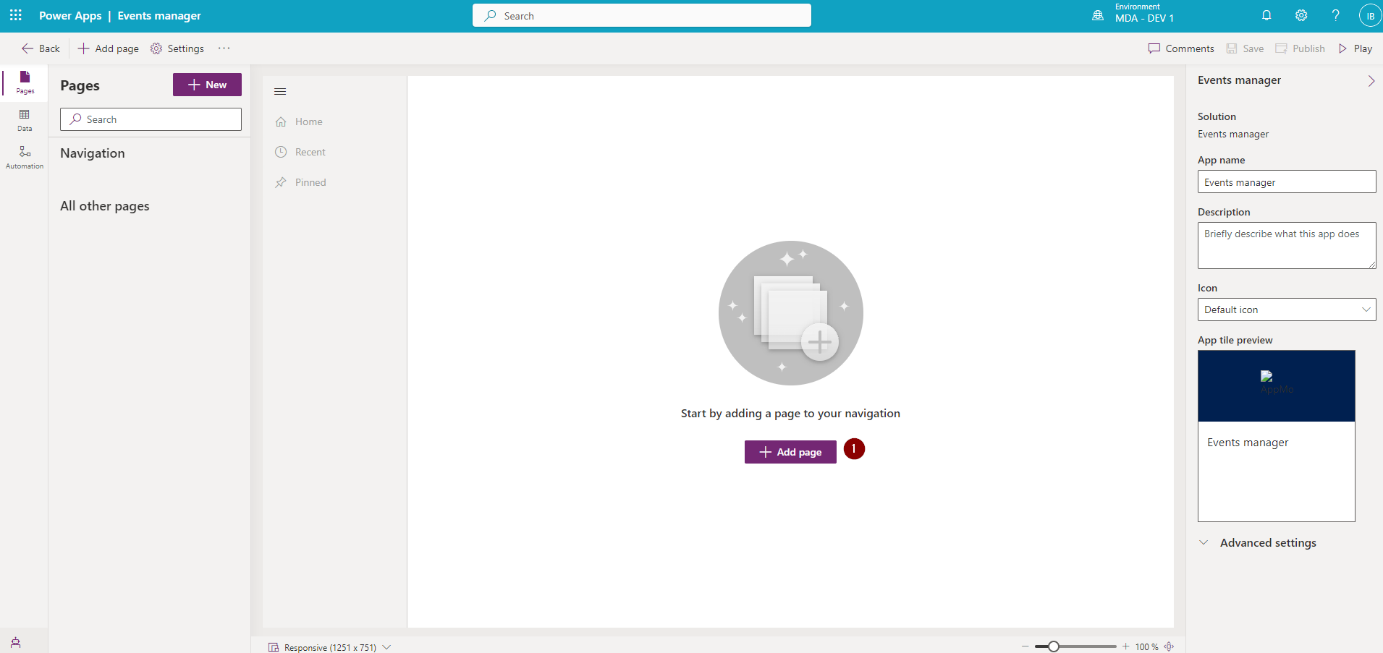
**CREATE AND EDIT NEW MODEL-DRIVEN APP**

In order to create new app, go to:



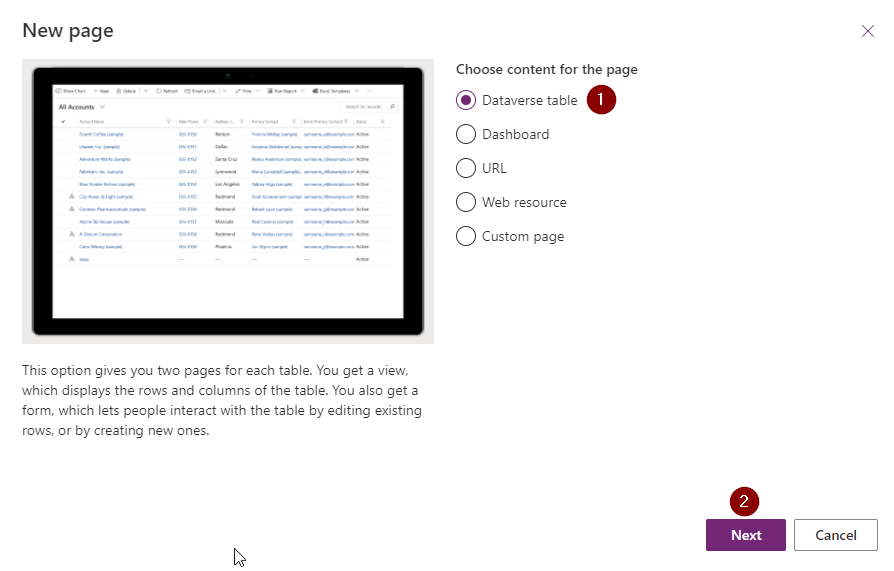
1. Objects.
2. App/Model-driven app.

Our app is a bit like a book, it has pages (that’s the only thing that these two have in common). You have to create the first page. In order to do so:

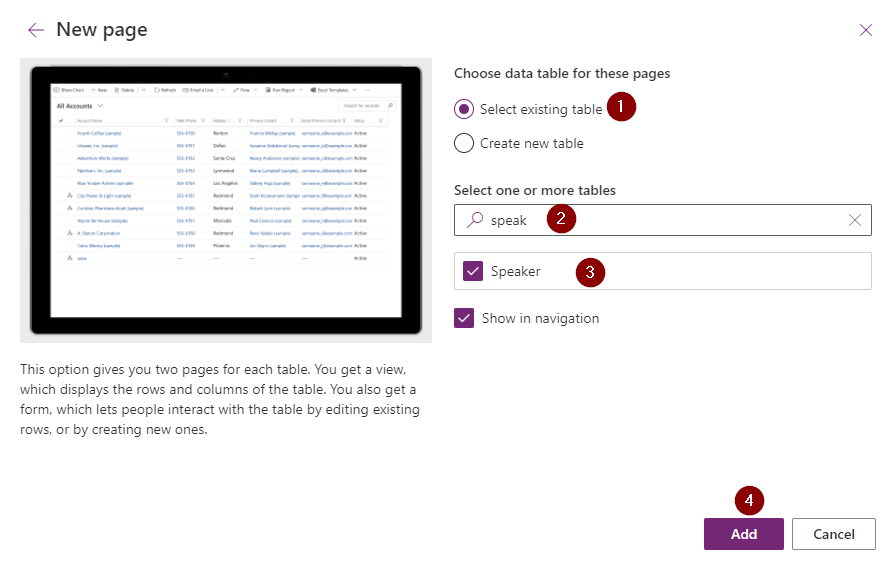


1. Click on the Add page button.

Then:

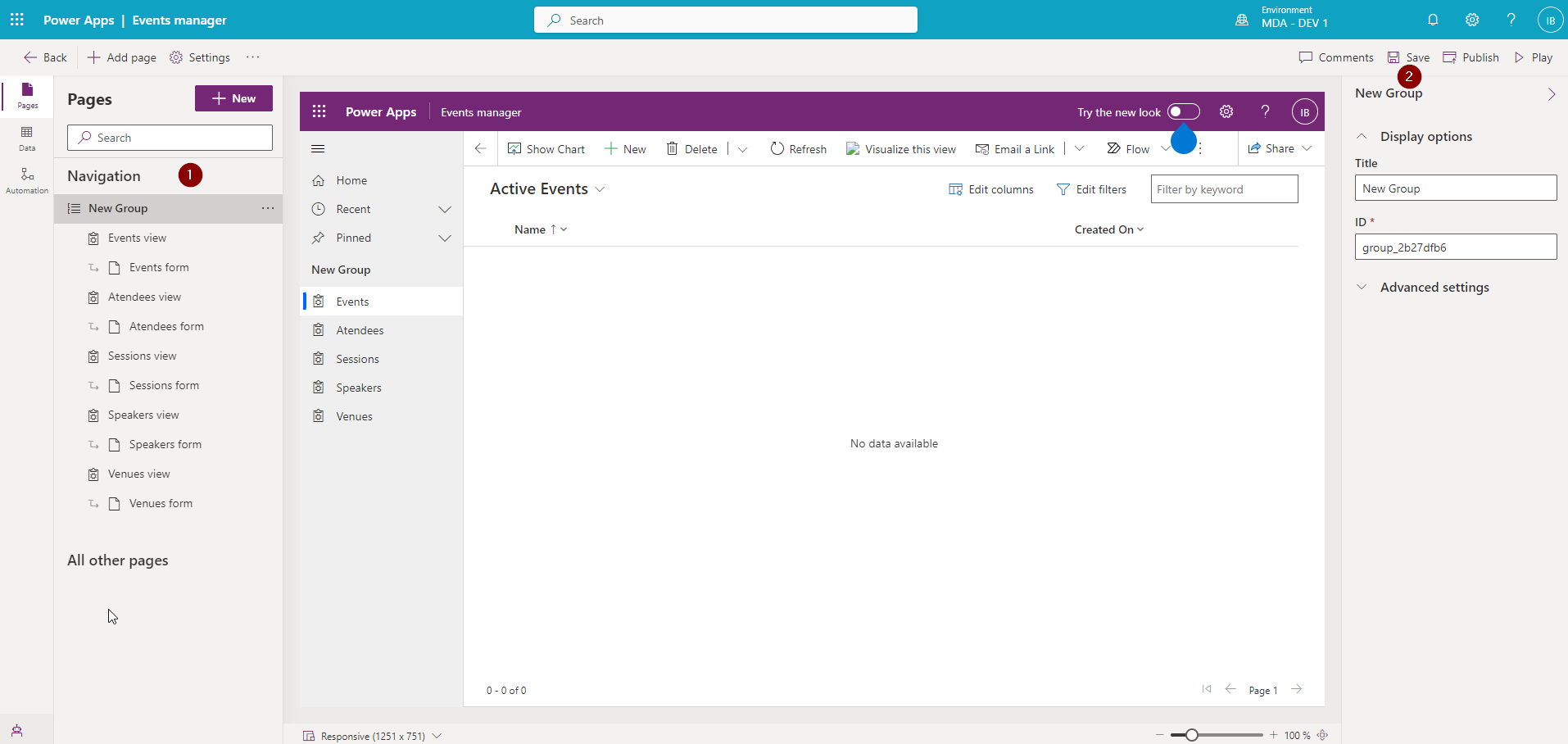


1. Select Dataverse table.
2. Click on Next.



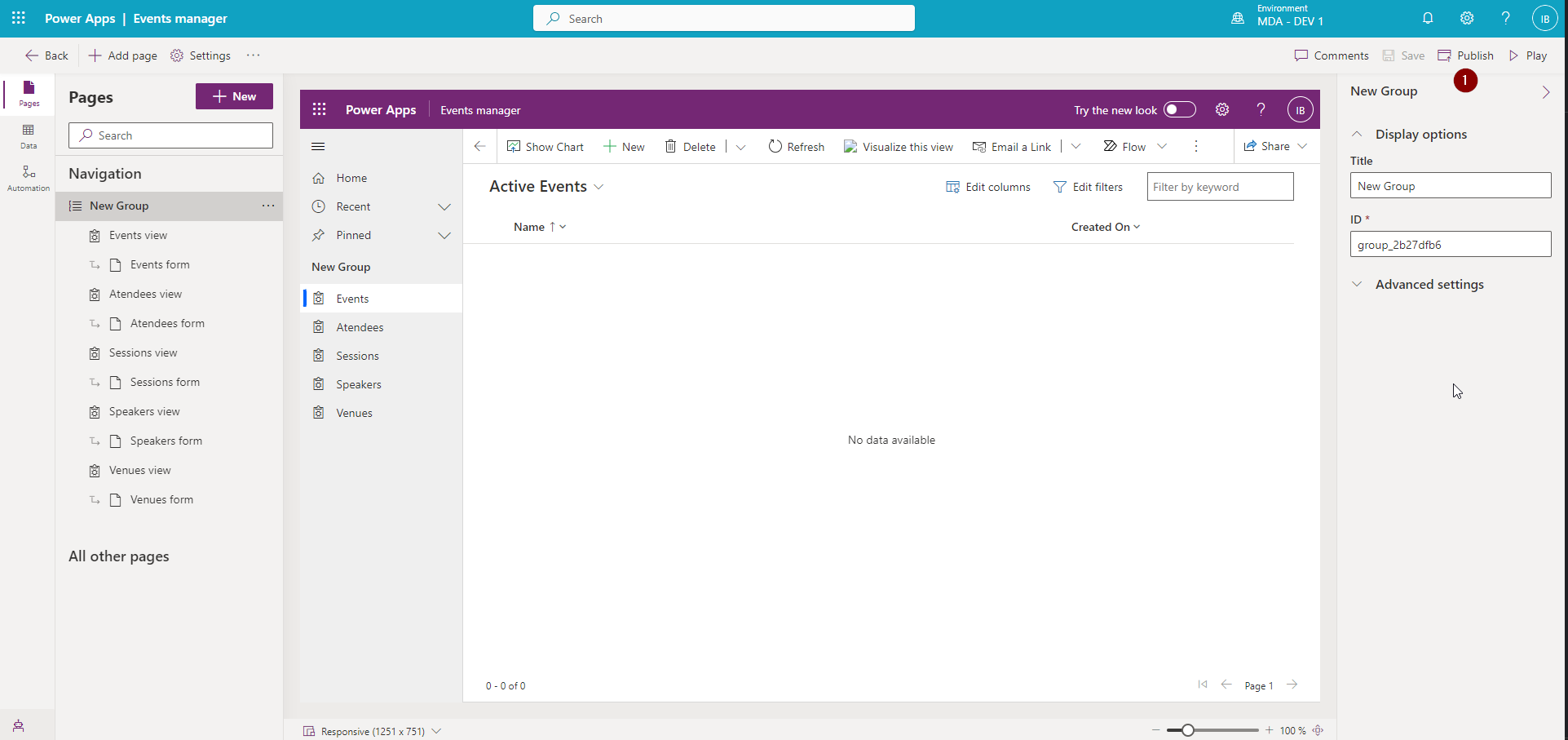
1. Select existing table.
2. Search for the right table.
3. Choose the right table or tables. Speaker in our case.
4. Click on Add.

The page will be added to the app.



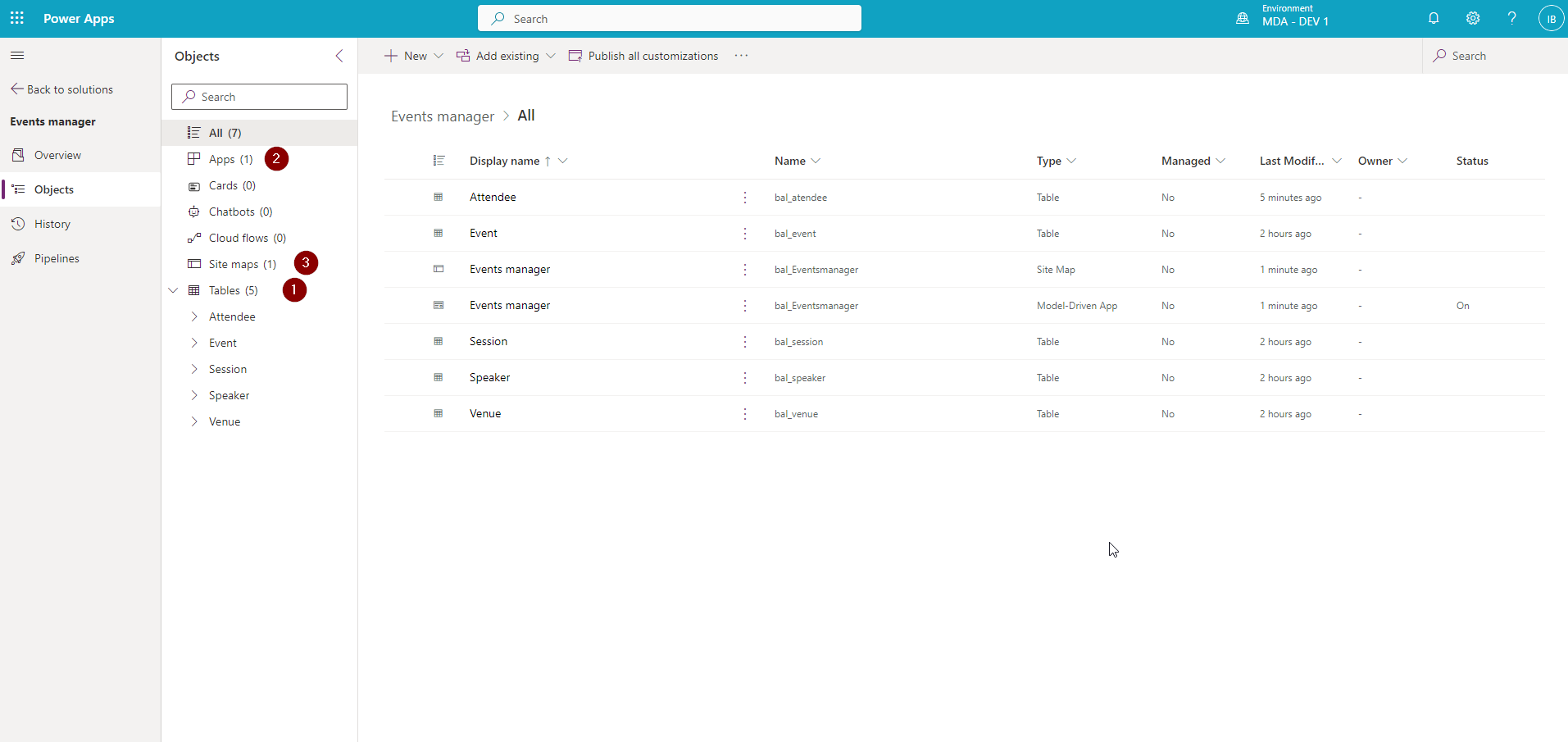
1. You can see it in the Navigation on the right.
2. If you’re satisfied with your Pages, click on Save button.

Once the app is saved:



1. Publish the app.

So far so good. What we’ve created till now:

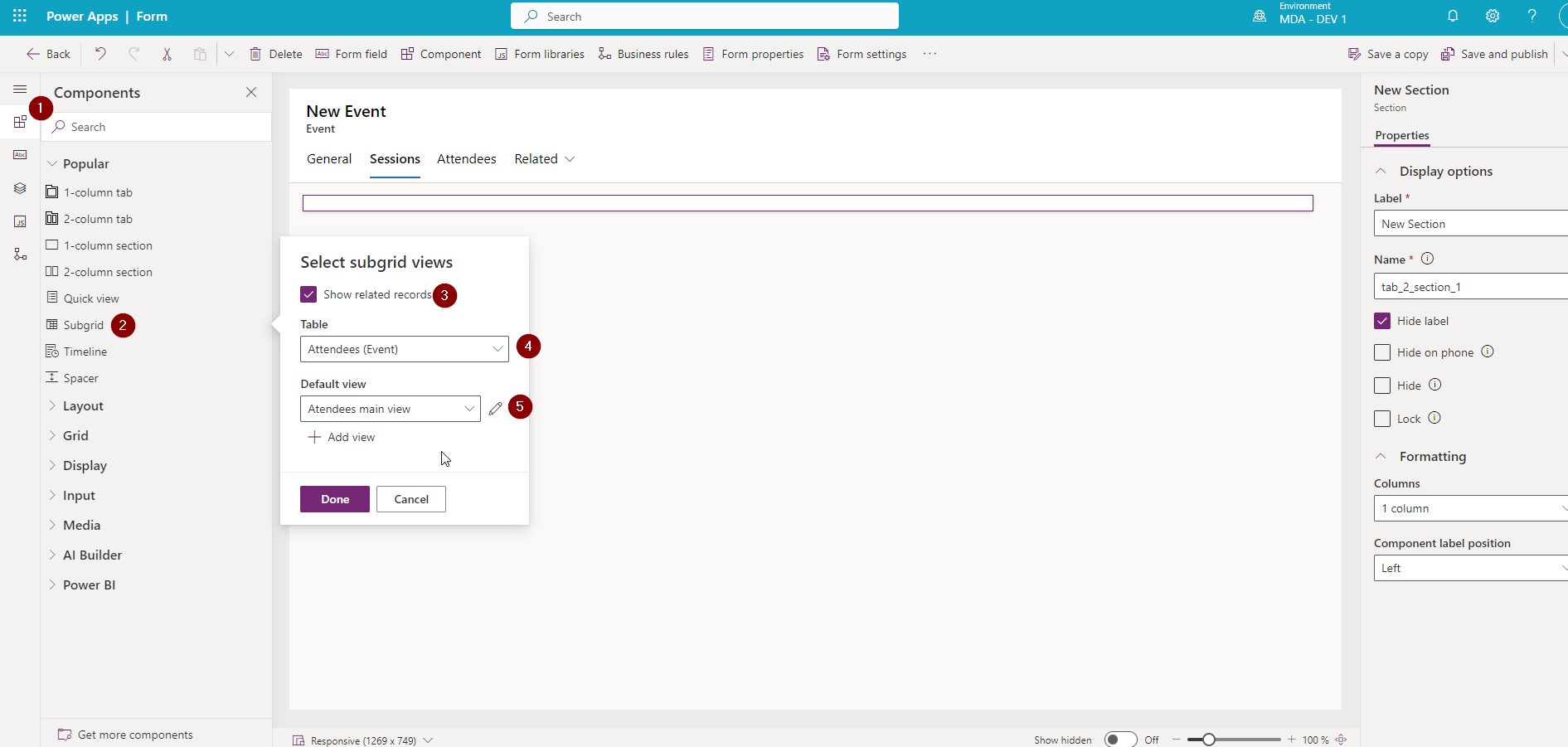


1. Tables. With all the columns, and all the forms, and all the views… We’re amazing.
2. App. Just. One. App.
3. Site map. Wait a minute! I didn’t create it! You are right. It was created for you once you created the app. Read more about it: [Create a model-driven app site map using the site map designer](https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/create-site-map-app).

We have something to build on. Let’s go back to our form and customize it a bit.

**FORM CUSTOMIZATIONS**

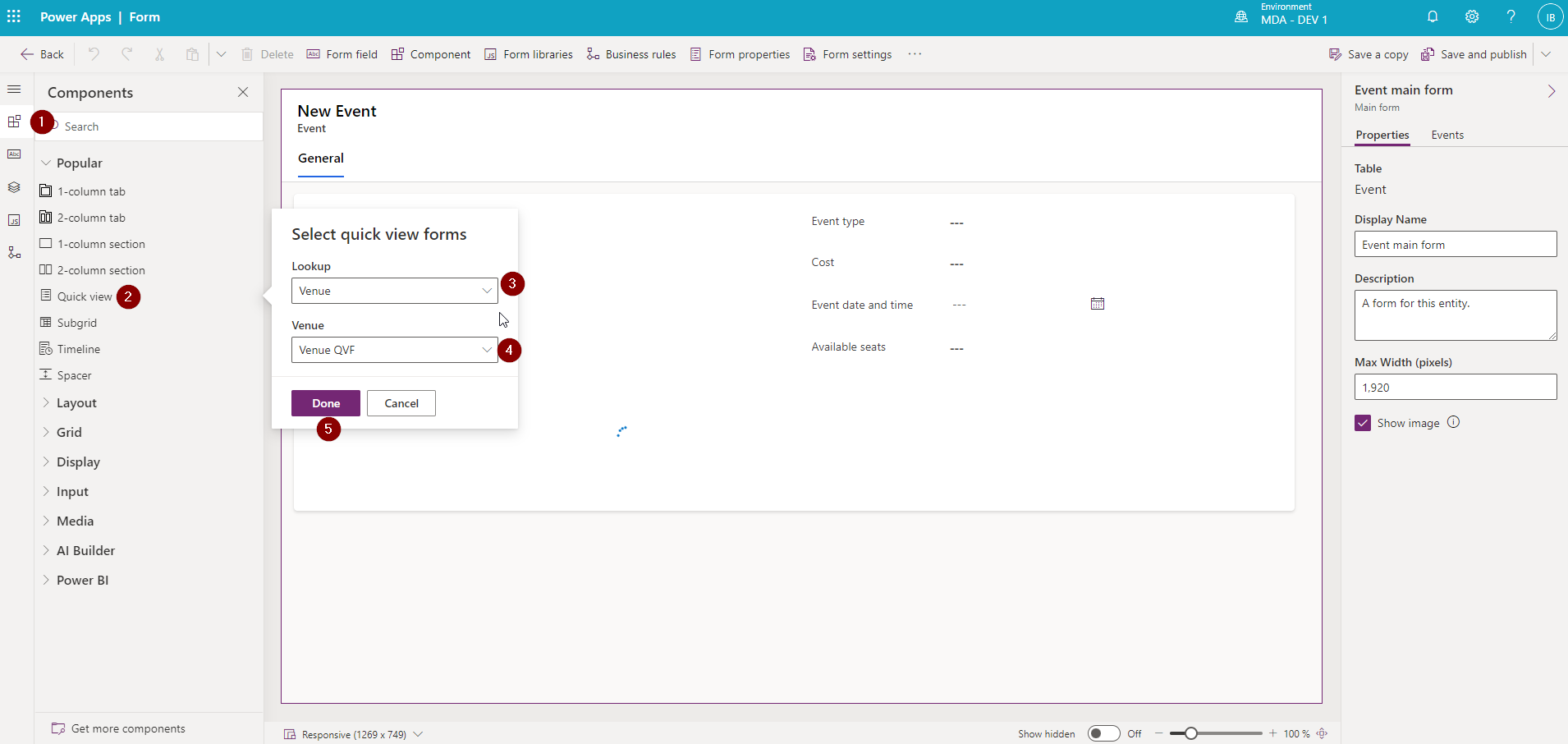
Let’s open the event form and add a subgrid. A subgrid is what we need if we want to show e.g. Attendees who signed up for the event.



1. Go to Components.
2. Select Subgrid.
3. Tick Show related records.
4. Select Attendees table.
5. Select the view and click on Done.

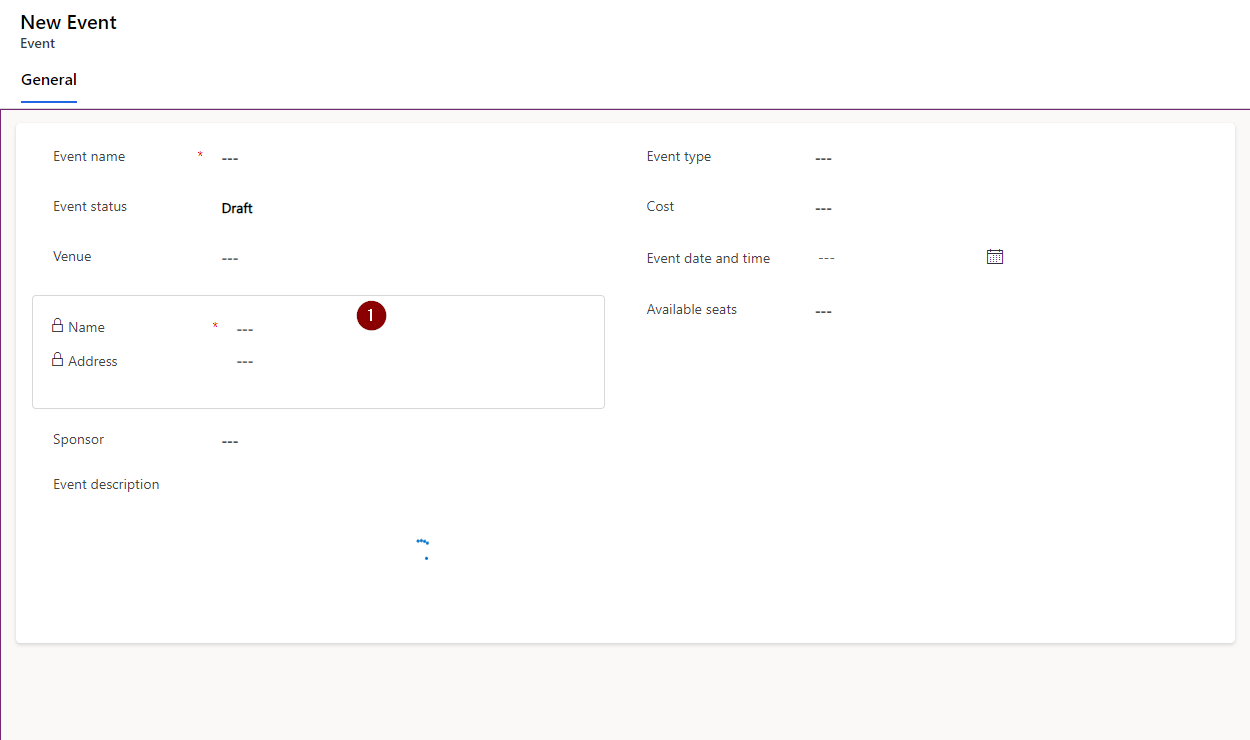
**ADD QUICK VIEW FORM TO MAIN FORM**

Quick view form is a form which allows you to show some additional information from the lookup column which is already on the form. You can edit it like we did with the Main form above, defining which columns should be visible. We’re gonna add some Venue details.



1. Go to Components
2. Select Quick view.
3. Select Lookup column.
4. Select QVF.
5. Click on Done.

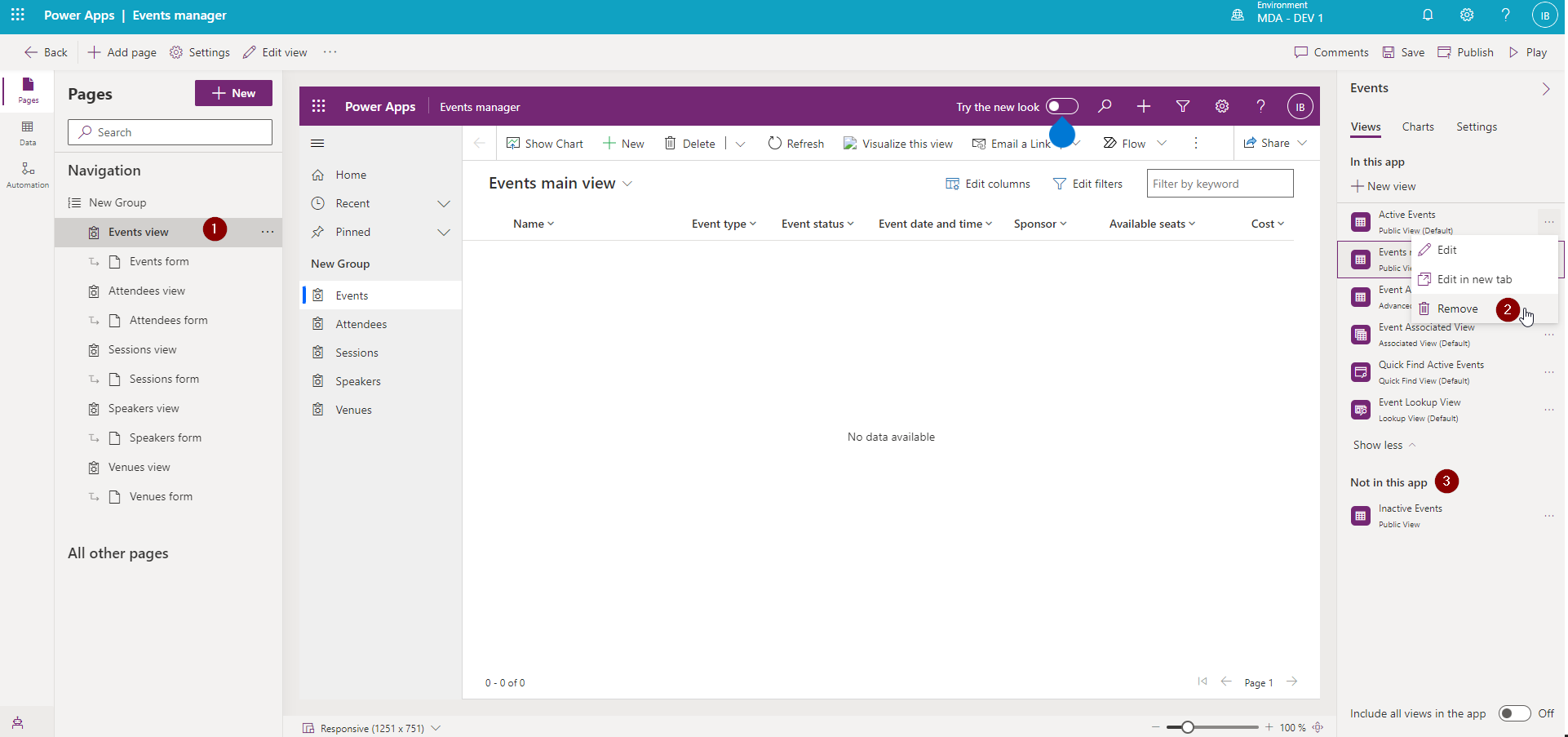
You can see the added QVF:



1. The QVF will be visible if the Lookup column is set.

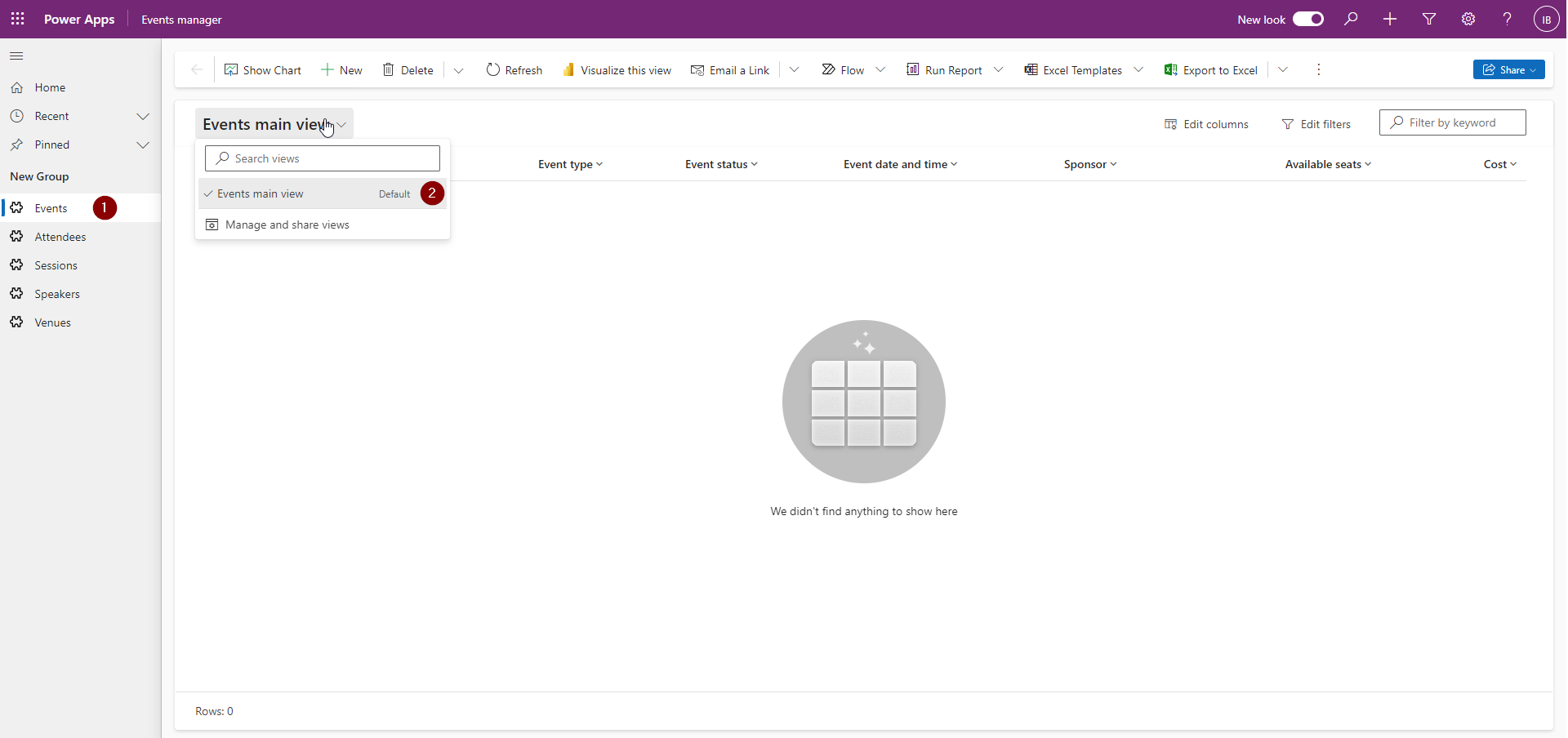
**REMOVE THE VIEW FROM THE APP**

If you don’t want particular view to be visible in the app:



1. Select the table views.
2. Click on the view you don’t want to see and select Remove.
3. See which views were removed from the app.

Now you can open the app and:

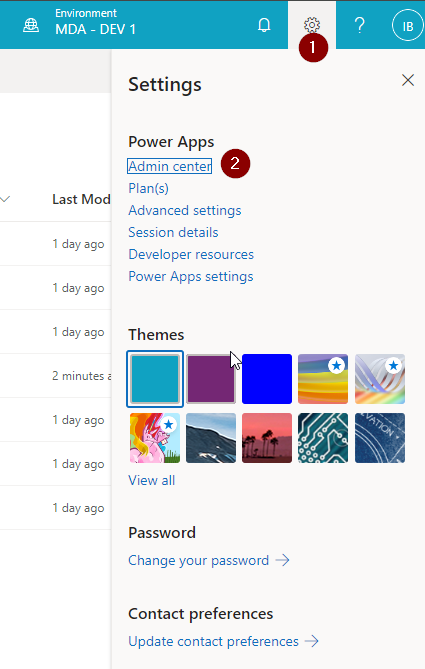


1. Go to the Events page.
2. See the views that weren’t removed from the app.

**OH ADMIN, MY ADMIN**

Admin Center is a center for Admins. Power Platform Admins, to be precise. Read more about it here: [Manage Microsoft Dataverse settings](https://learn.microsoft.com/en-us/power-platform/admin/admin-settings).

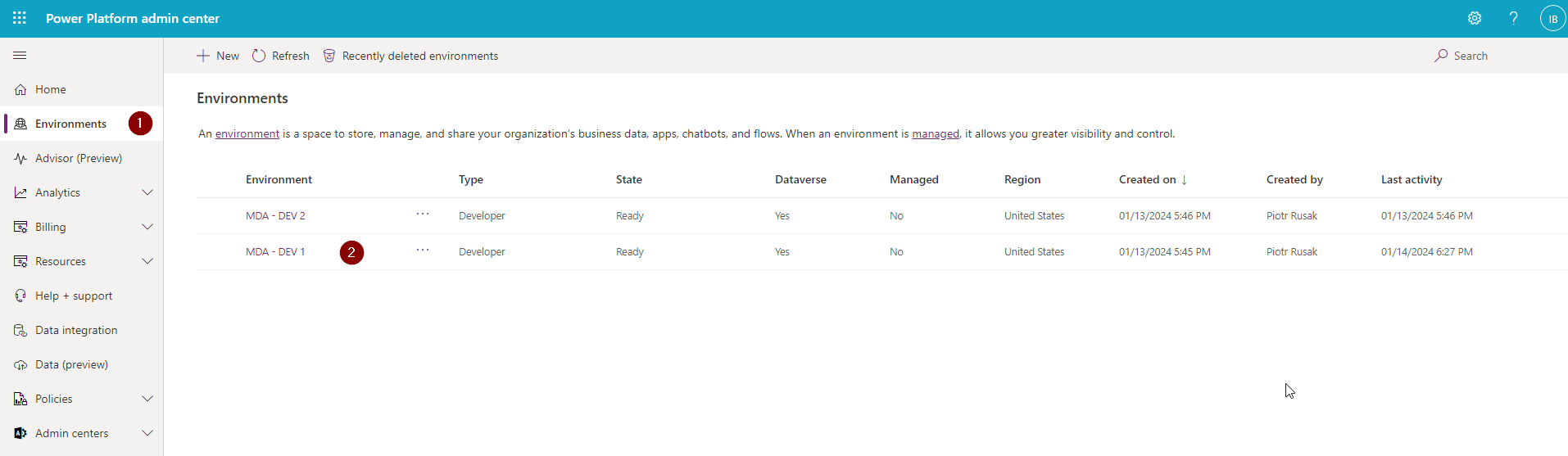
To acces it you need to open the environment and:



1. Click on Setting icon.
2. Click on Admin center.

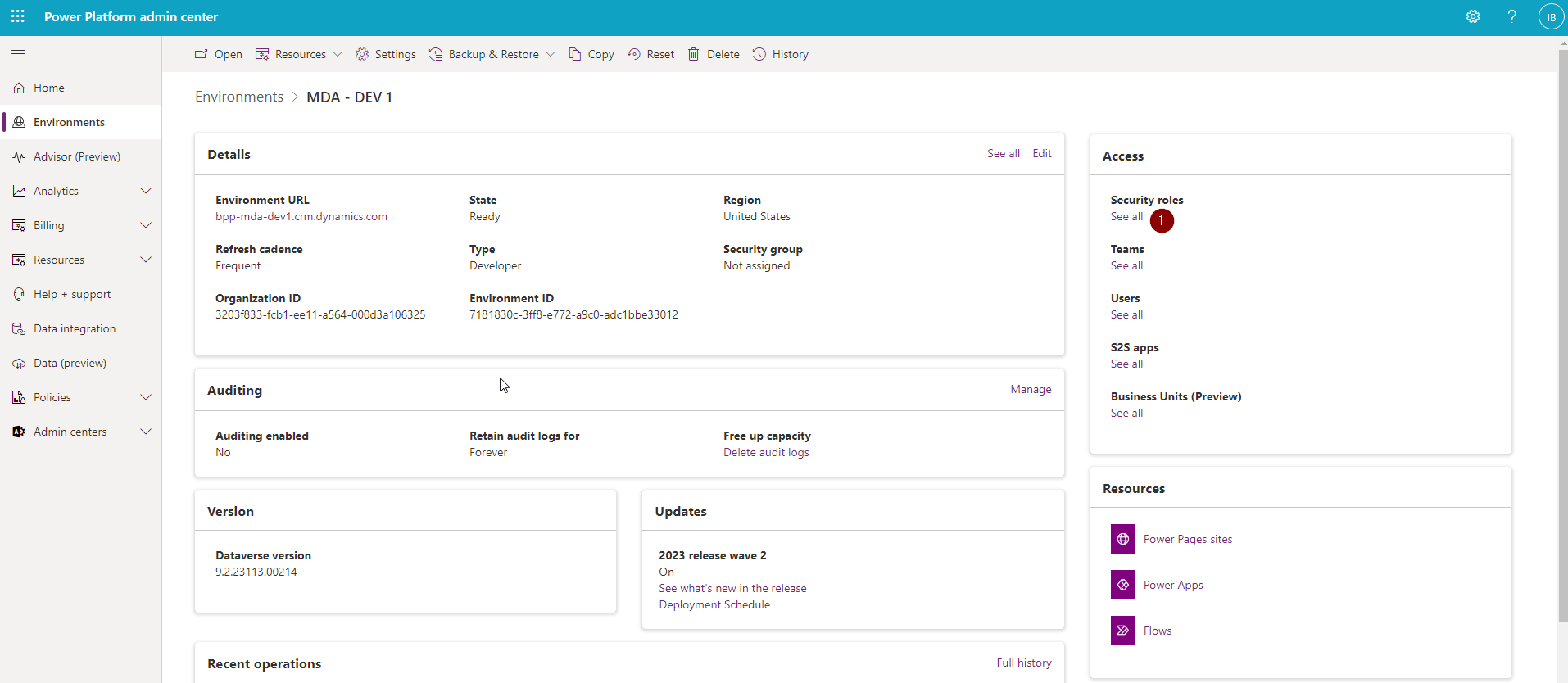
**SECURITY ROLE**

The Admin center will open. At first, we’re gonna take a look at Security roles.



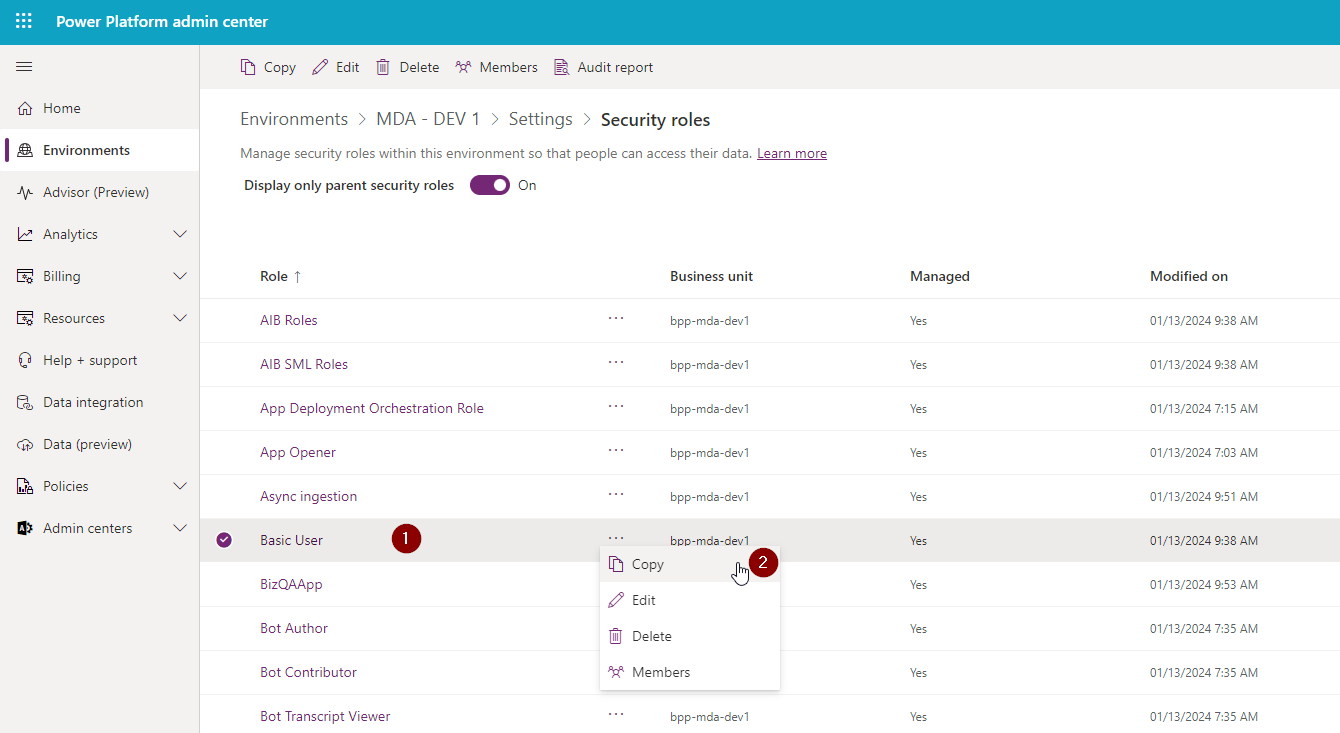
1. Click on Environments
2. Select the right Environment (not the one on the right – the right for you).

You will see an overview of Environment options.



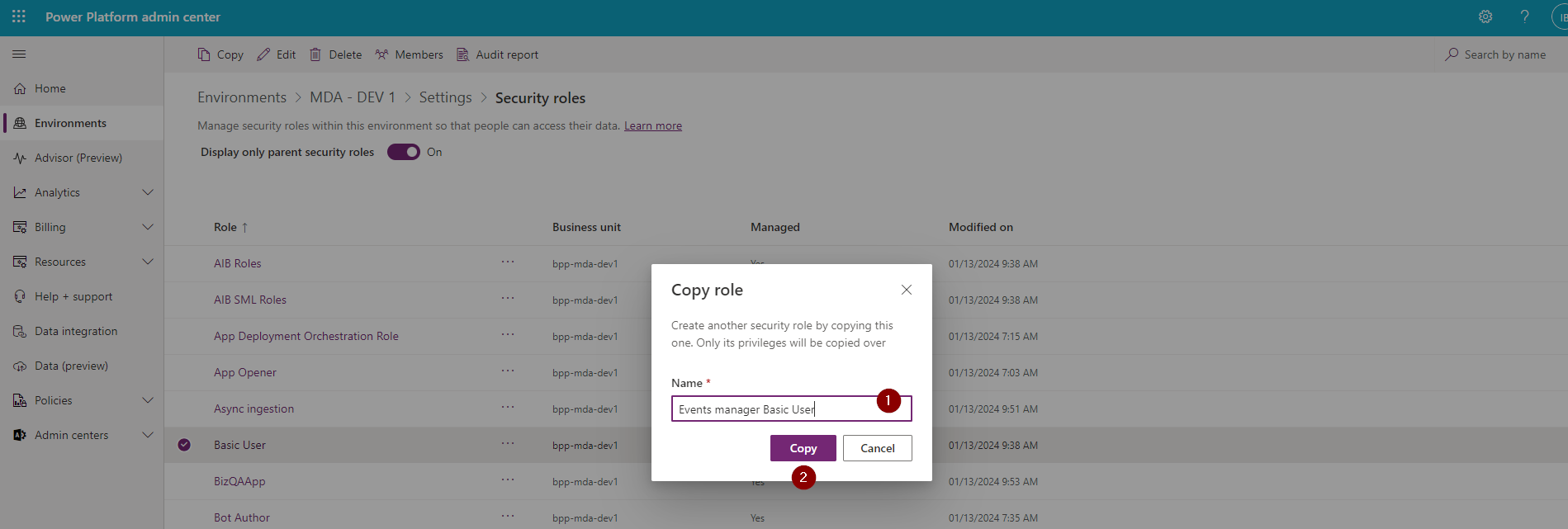
1. Click on See all below Security roles.

You will see all the security roles available in the environment.



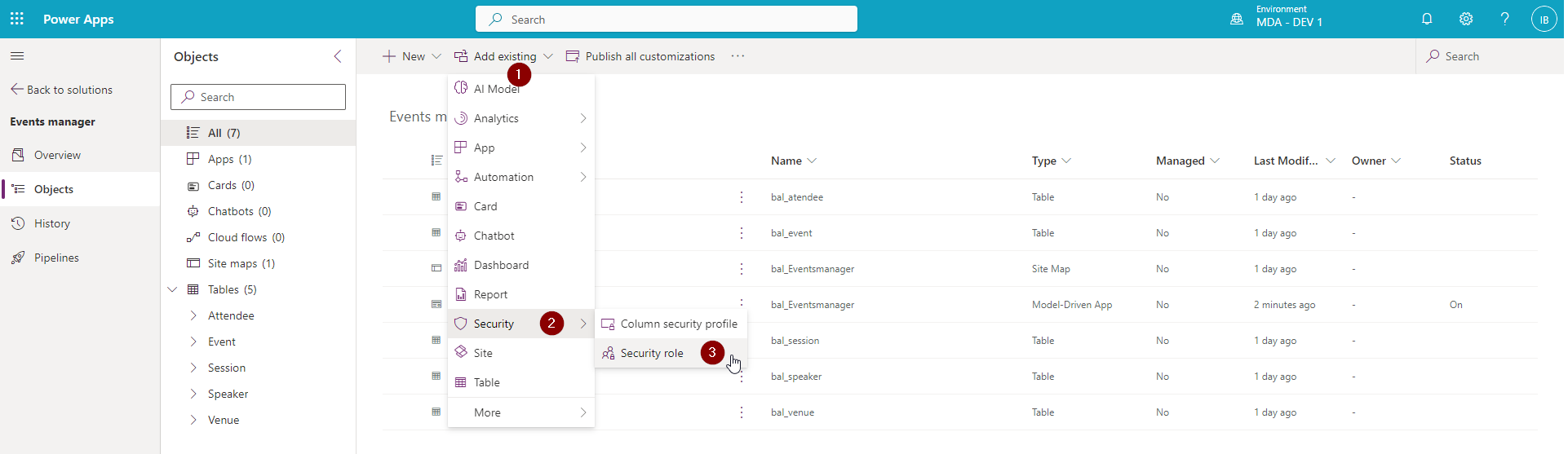
1. Click on three dots next to Basic User.
2. Click on the Copy button.

It’s good practice to create a separate Basic User role for the specific solution, which you can further customize.

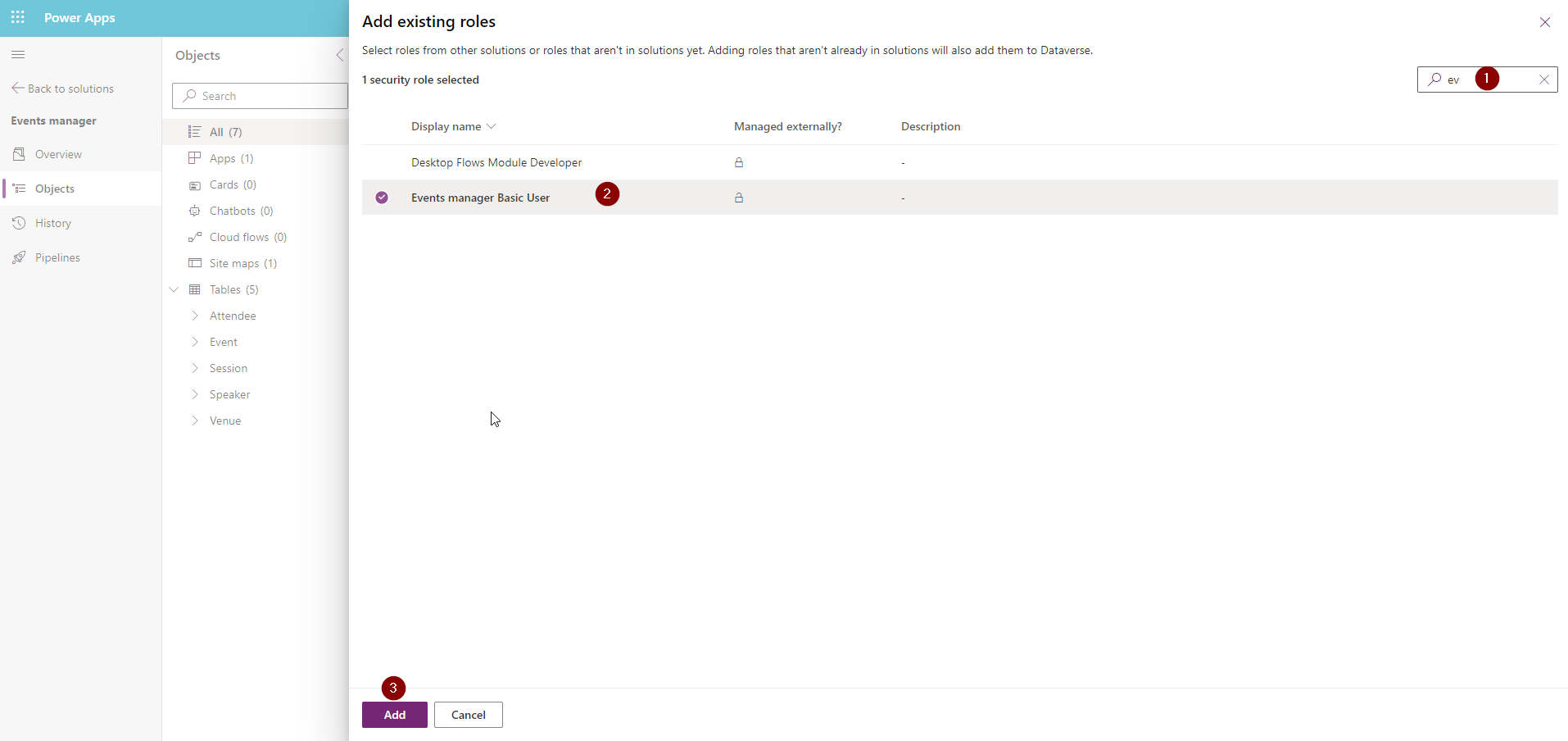


1. Write the name (I usually use <Solution name> Basic User convention).
2. Click on Copy.

Now having the copy, we want to add it to our solution. Go to solution:

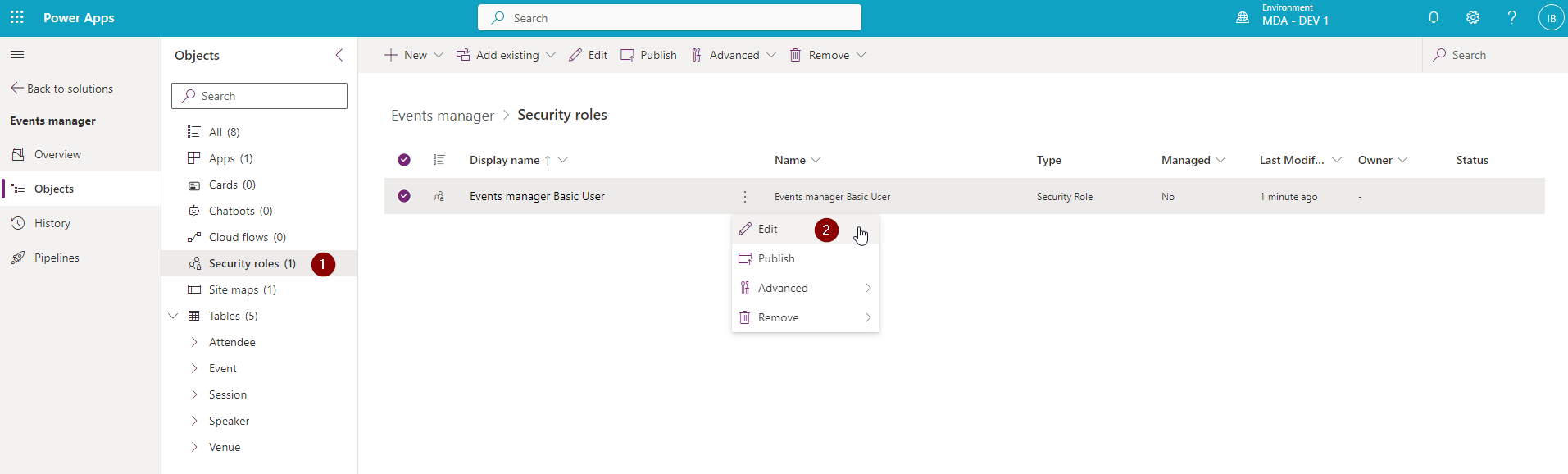


1. Select Add Existing.
2. Security.
3. Security role.



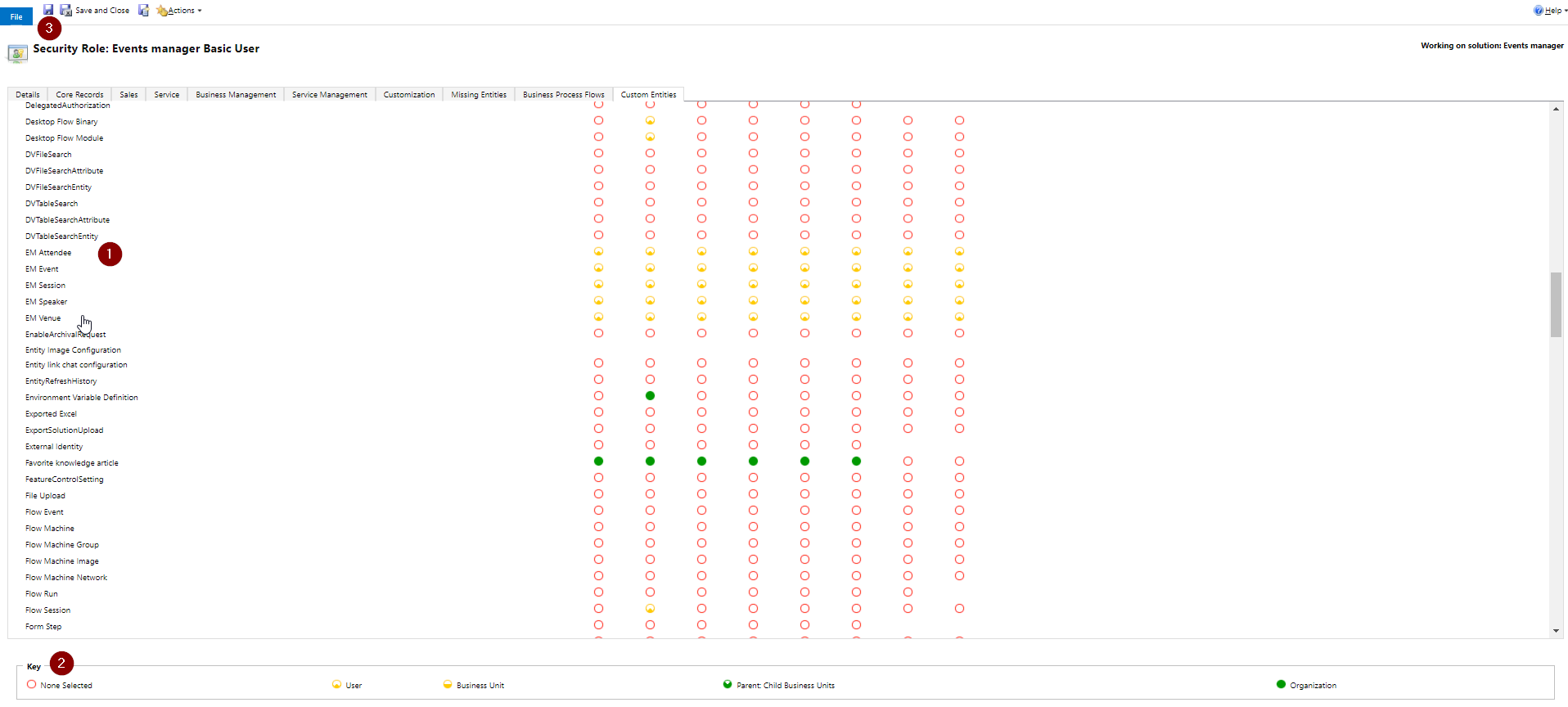
1. Search for the Events manager Basic User role.
2. Select it.
3. Click Add.

The security role was added to the solution. Now we can go to:



1. Security roles.
2. Click on three dots next to the role and select Edit.

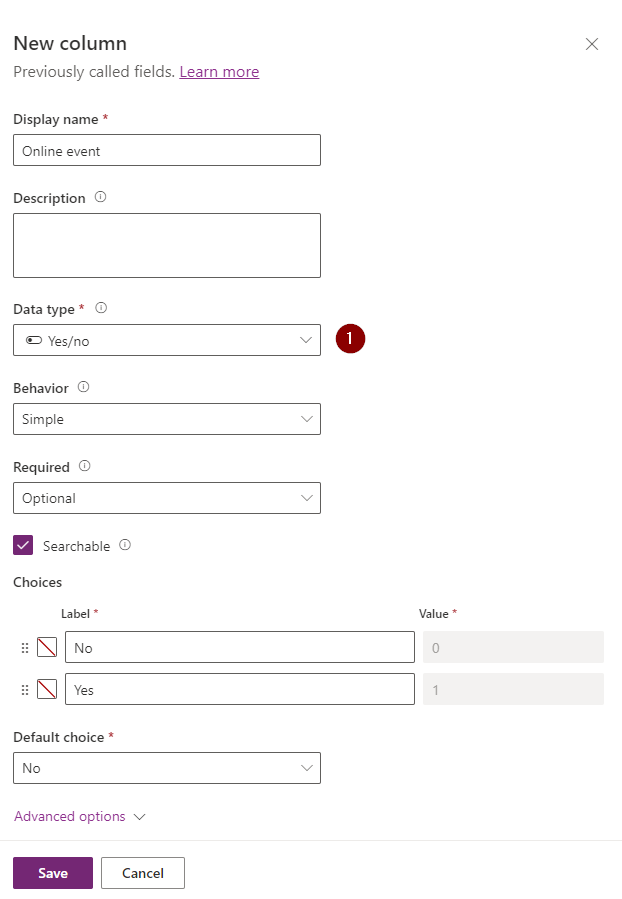
Now you can define to which tables it has permissions.



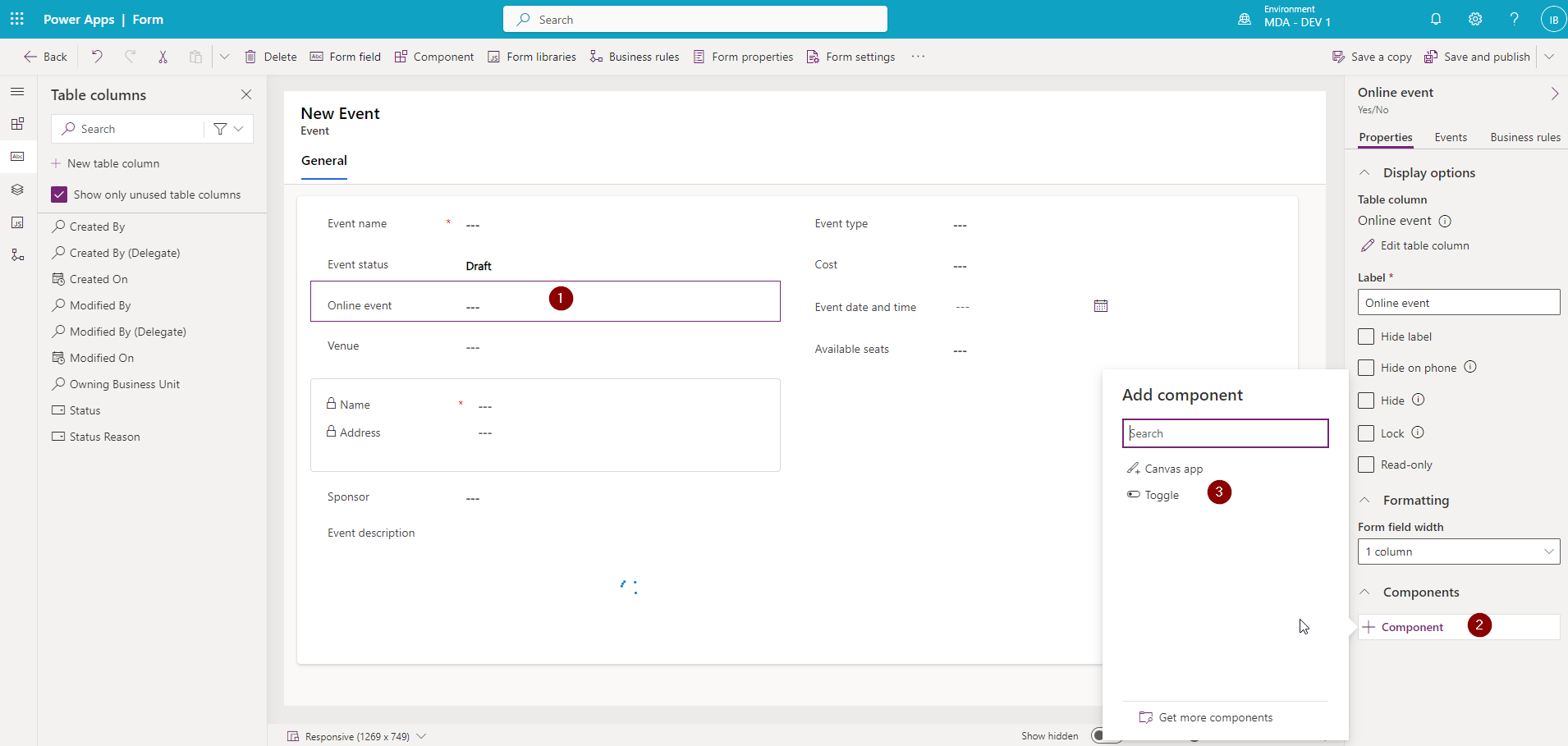
1. Find the tables for which you want to set a permission.
2. Define proper permission level.
3. Save security role.

**ADD YES/NO COLUMN AND SHOW IT AS TOGGLE ON FORM**

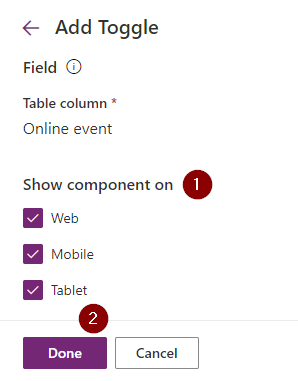
Add new column of Yes/No type.



Edit Form and add it to it.



1. Click on the column on form.
2. Select Component.
3. Add Toggle component.



1. Define on which devic the toggle component should be visible.
2. Click on Done.

Now you’ve got yourself a toggle. It ain’t much, but it’s an honest component.

**POPULATE FEW ATTENDEES USING MS FORMS INTEGRATION**

Go to <https://forms.office.com/Pages/DesignPageV2.aspx?origin=shell> using same tenant where you are building whole solution.

Create simple MS Form survey with 6 Text Fields:

First Name (Mandatory)

Last Name (Mandatory)

Email address (Mandatory)

Company

Job Title

Phone Number

A screenshot of a computer

Description automatically generated

Set unique Theme color for the MS Form

Publish MS Form and preserve Title of the Event + Short URL + MS Form Id

Short URL

A screenshot of a computer

Description automatically generated

MS Form Id

A screenshot of a computer

Description automatically generated

Duplicate Form few times (at least 2 times)

For duplicated MS Forms specify different Title and Theme Color (just to differentiate forms)

For all duplicated forms preserve Title of the Event + Short URL + MS Form Id

**CREATING MANUALLY EVENTS IN DATAVERSE**

Go to MDA App and open Events section

Create Events corresponding to created MS Forms (at least 3)

A screenshot of a computer

Description automatically generated

**PLAYING WITH POWER AUTOMATE**

Go to Events manager solution

Go and create 4 Connection References to following Services:

* Microsoft Dataverse
* Microsoft Forms
* Office 365 Outlook
* Power Automate Management

Inside Solution create first Flow – Instant with Title:

[DEMO-1] Process Response (Child Flow) [Instant]

and Trigger

A screenshot of a computer

Description automatically generated

With 6 Input Parameters (single line of text)

Event

FirstName

LastName

Email

Company

JobTitle

PhoneNumber

Using **List rows** action from Dataverse find Event that equals to Event (input property)

Use FetchXML below:  
  
<fetch top='1'>

<entity name='bal\_event'>

<attribute name='bal\_availableseats' />

<attribute name='bal\_cost' />

<attribute name='bal\_eventdateandtime' />

<attribute name='bal\_eventdescription' />

<attribute name='bal\_eventid' />

<attribute name='bal\_eventstatus' />

<attribute name='bal\_eventtype' />

<attribute name='bal\_name' />

<attribute name='bal\_sponsor' />

<attribute name='bal\_venue' />

<filter>

<condition attribute='bal\_name' operator='eq' value='@{triggerBody()['text']}' />

</filter>

</entity>

</fetch>

Rename action to **FindEvent**

Drop Compose Action and paste code below to this action (save flow and make sure Save works)

@{if(greaterOrEquals(length(outputs('FindEvent')?['body/value']),1), concat('bal\_events(',outputs('FindEvent')?['body/value'][0]['bal\_eventid'], ')'), null)}

Add new row to Attendees table (use Dataverse **Add new row** action)

Fill in columns like on screen below (ask host in case of any issues)

Rename this action to CreateAttendee

A screenshot of a computer

Description automatically generated

Place Respond to Power Apps action with following response parameter

A screenshot of a computer

Description automatically generated

Save flow and get back to this flow Details page

Setup Run only users for this flow (this action is required for all Child Flows)

A screenshot of a computer

Description automatically generated

Create solution aware flow that listens to Responsed to one of your MS Forms (pick one)

Trigger – When new response is submitted:

A screenshot of a computer

Description automatically generated

Then read details of response submitted (DO NOT hardcode MS Form Name – you can reference this from trigger input)  
@trigger()?['inputs']?['parameters']?['form\_id']

A screenshot of a computer program

Description automatically generated

Place Compose action and prepare JSON object with following values:

{

"Event": "Baltic Power Platform Meetup - February 2024",

"FirstName": "{SELECT-FROM-FormResponse}",

"LastName": "{SELECT-FROM-FormResponse}",

"Email": "{SELECT-FROM-FormResponse}",

"Company": "{SELECT-FROM-FormResponse}",

"JobTitle": "{SELECT-FROM-FormResponse}",

"PhoneNumber": "{SELECT-FROM-FormResponse}"

}

A screenshot of a computer

Description automatically generated

Use Parse JSON action to pare output of AttendeeData to give Hints in next part of the flow

Use JSON Object notation as Sample Payload to generate JSON Schema

A screenshot of a computer

Description automatically generated

Call Child Flow that was created before – match all parameters by name (should be easy)

A screenshot of a computer

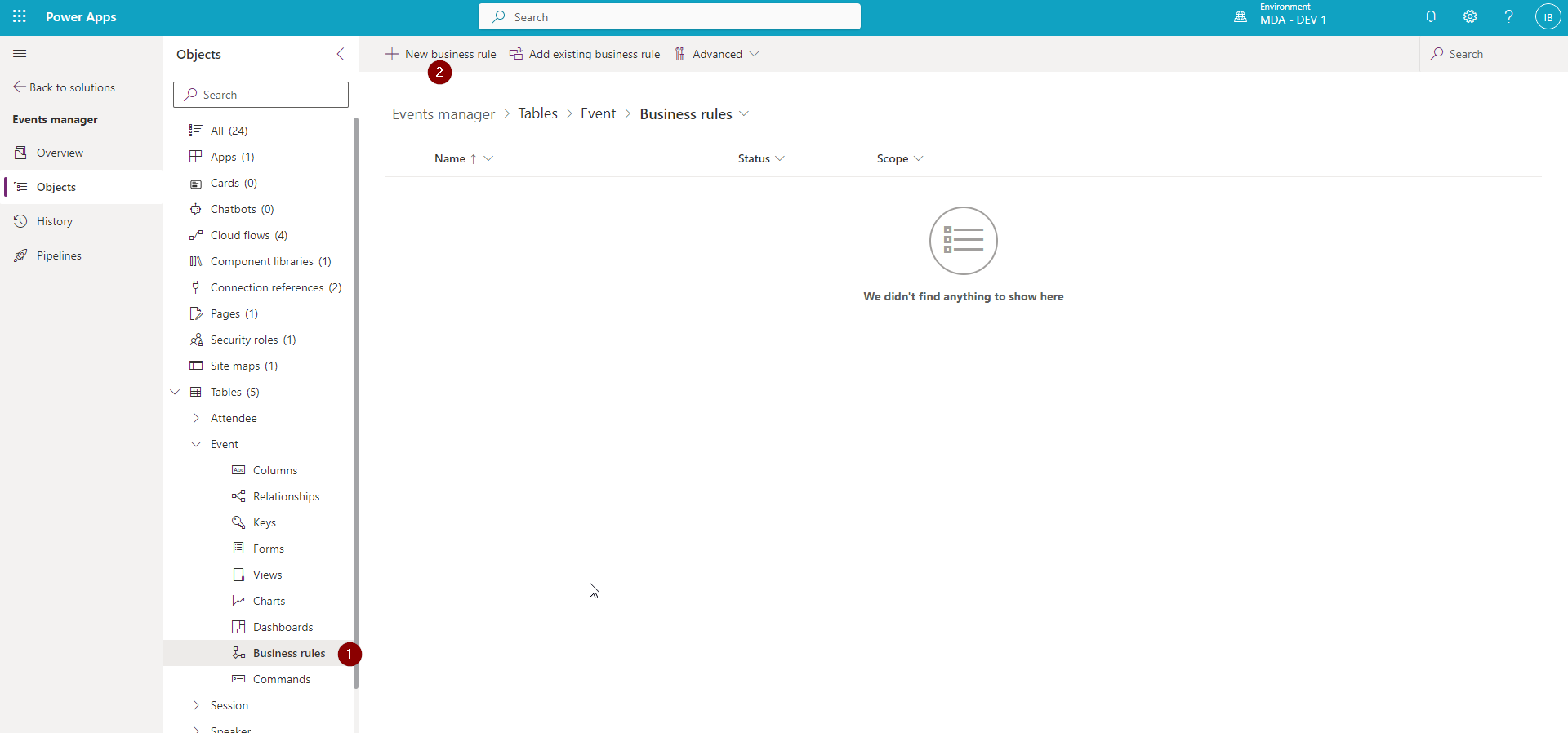
Description automatically generated

Bravo – you have just created automation that gives you ability to add Attendees to particular event.

But wait – we created MS Forms for few other events – should we manually create flows that will listen to those other MS Forms ?? Of course – not (we will present 2 nifty ways of saving your precious time)

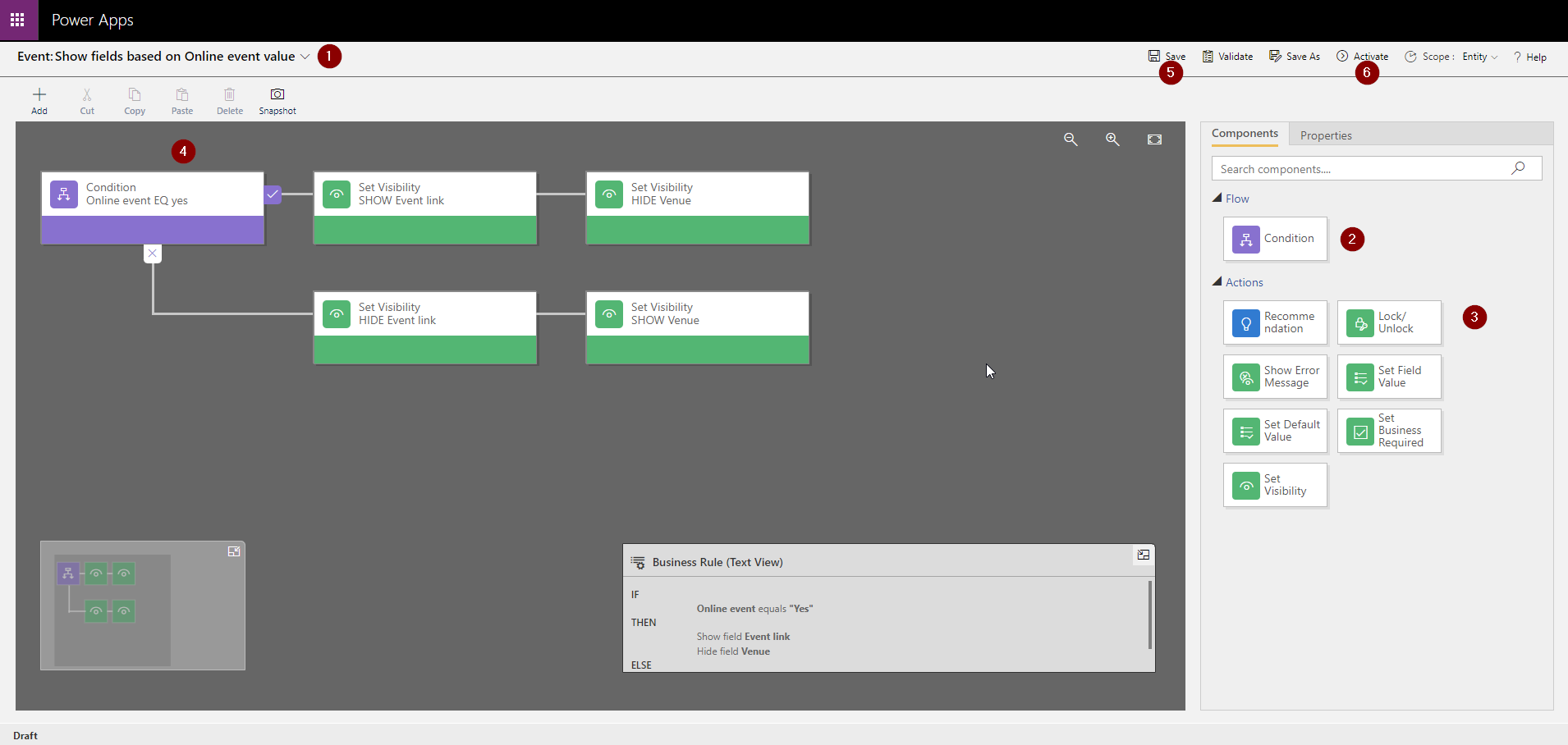
**CREATE BUSINESS RULE**

Business rules are useful if you want to e.g. based on a specific condition define fields visibility or values on form. Go to solution.



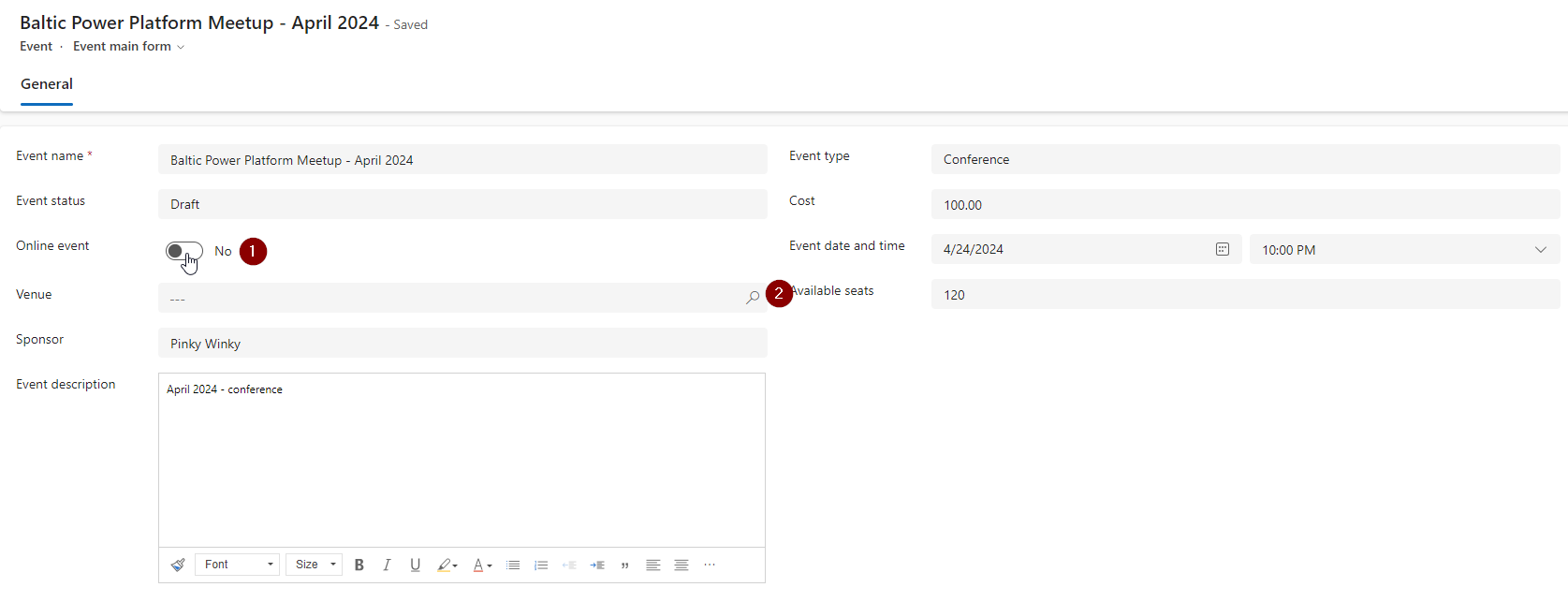
1. Click on Business rules.
2. Click on New business rule.

In our example we will create a simple business rule which shows and hides fields based on the Online event field value.

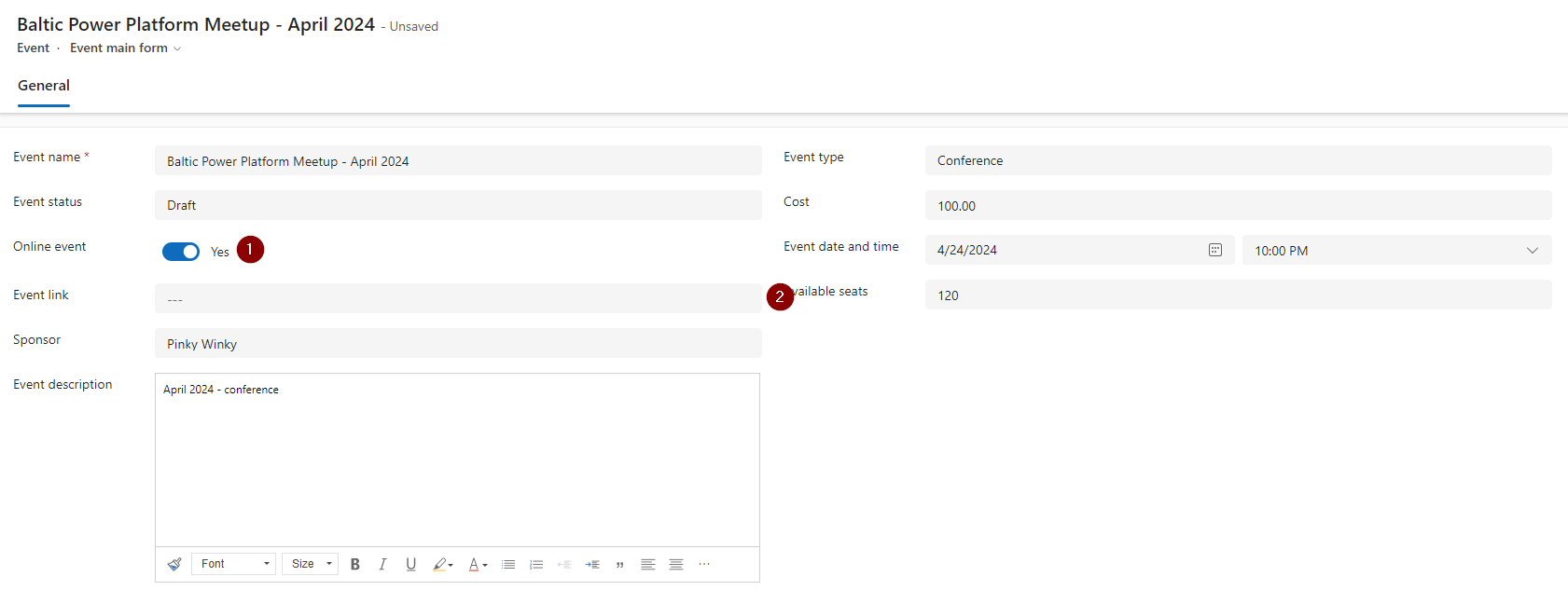


1. Set business rule name.
2. Add Condition.
3. Add Actions.
4. Save business rule.
5. Activate it.

Now the new business rule will apply to the form.



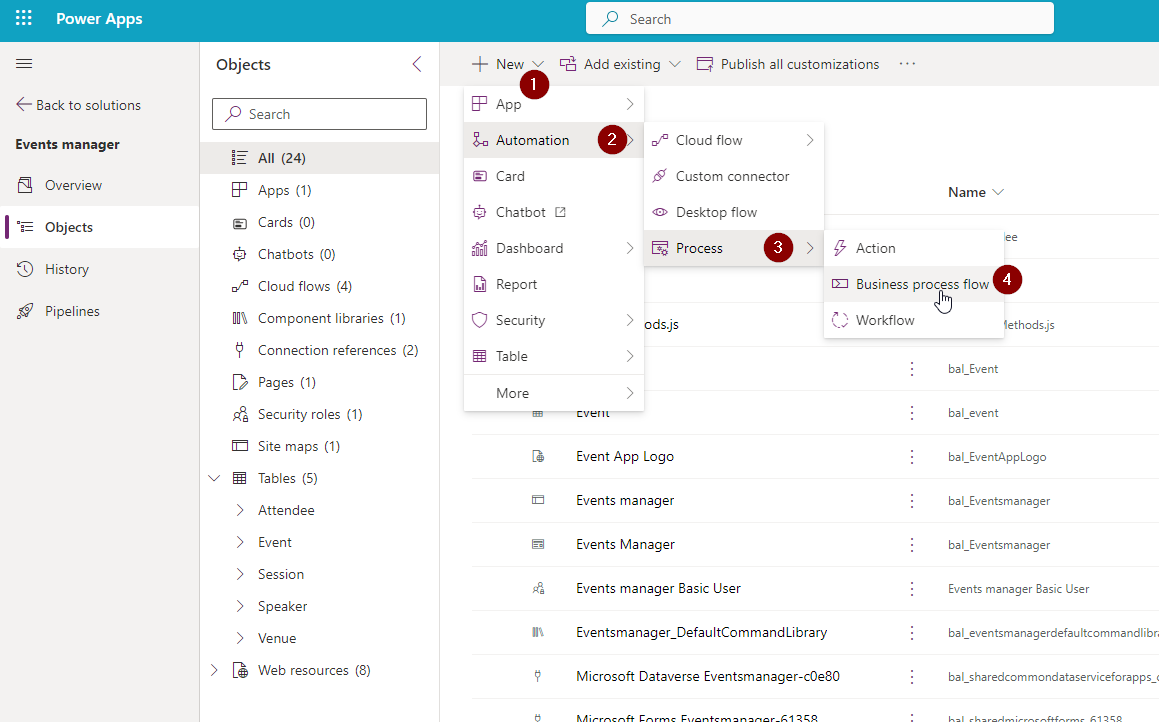
1. Set Online event (the toggle field!) to No.
2. Venue lookup field is visible.



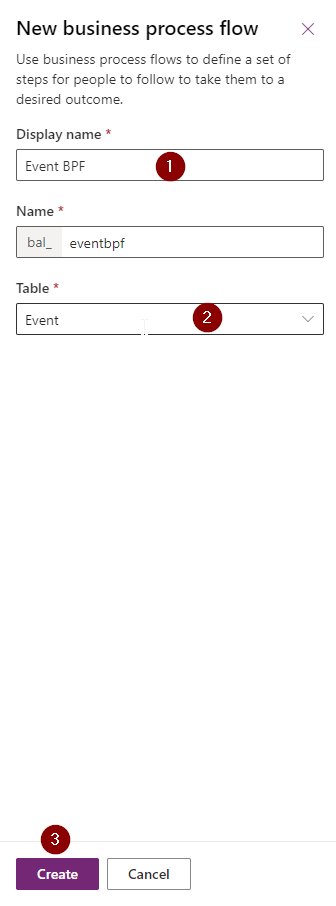
1. Set Online field to Yes.
2. Event link field is visible.

**BUSINESS PROCESS FLOW**

Business process flow (BPF) allows us to create steps e.g. which should be followed when filling in the data. We will create a simple BPF for the Event table.

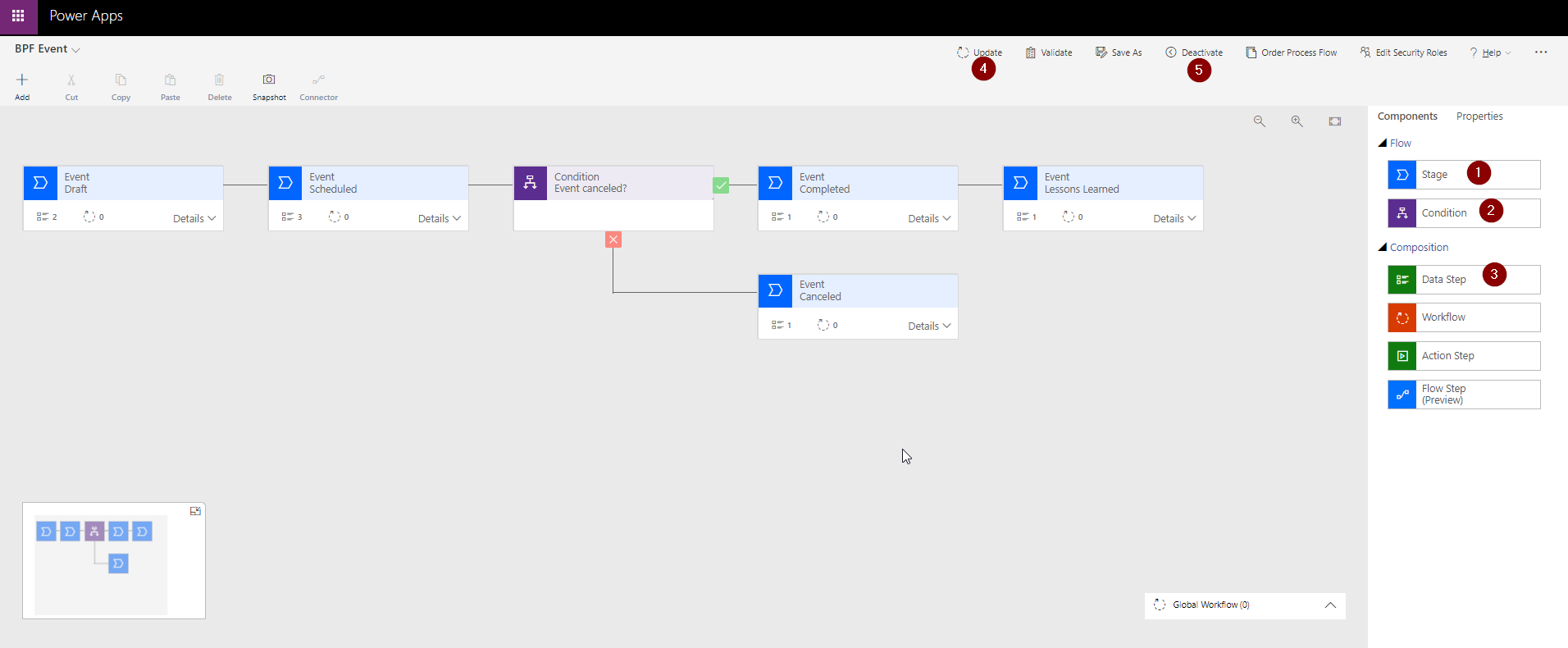


1. Go to solution and select New.
2. Automation.
3. Process.
4. Business process flow.



1. Give your process a name.
2. Connect it to a specific table.
3. Click on the Create button.

Once the BPF is created, you can edit it:



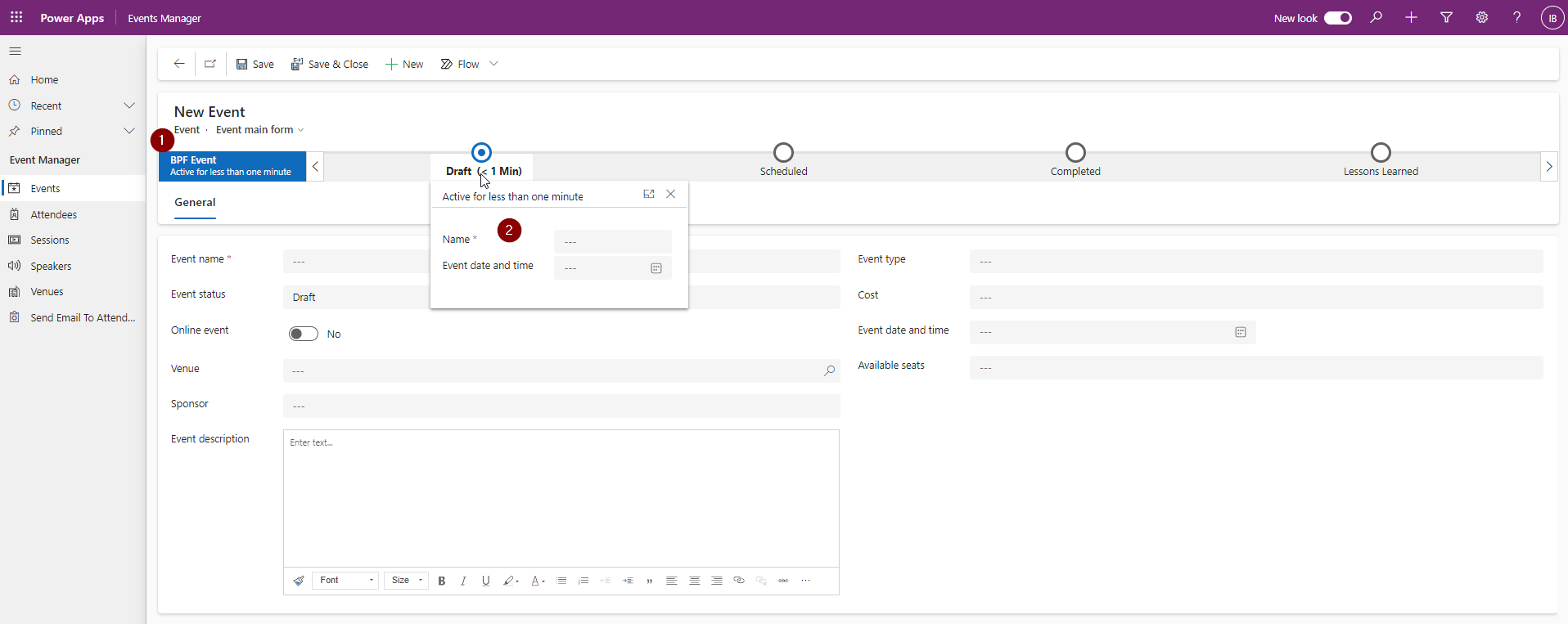
1. Add stages.
2. Add conditions.
3. Add data steps to stages (in our example).

We created a quite simple BPF with 5 stages: Draft, Scheduled, Completed, Lessons Learned and Canceled. There is one condition which influences the following stages.

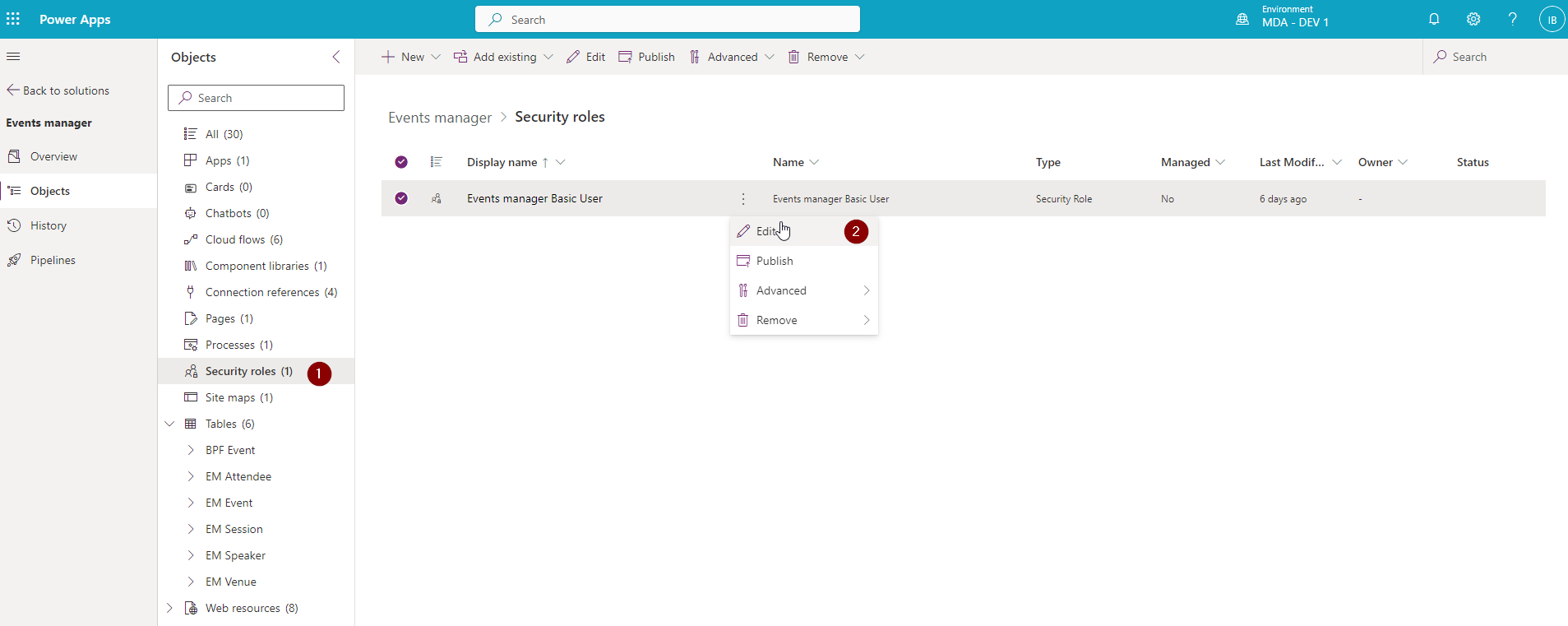
Once we create our BPF logic we can:

1. Update it.
2. Activate or Deactivate it. Only the Activated BPF will be available on the form.

And that’s how the BPF presents on Event form:



1. We can see the BPF with various stages.
2. For each stage we can see the predefined data step with column that need to be filled in for that stage.



**CUSTOMIZING MDA APPLICATION – MODERN COMMANDING / CUSTOM PAGES / POWER AUTOMATE INTEGRATION**

In next 2 parts we will present how to extend MDA using:

* Custom Modern Commanding buttons
* Custom UI – Power Apps Custom Pages
* Custom Automations – multiple ways of executing Power Automate directly from Model Driven Apps

**HOW TO CALL POWER AUTOMATE FLOWS FROM MDA – METHODS COMPARISON**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Method** | **Sync** | **Async** | **Notes** |
| 1 | Using trigger from Dataverse (Legacy) – “When a row is selected” | No | Yes | This has OOTB method of display and integration |
| 2 | Calling Power Automate flow from Canvas App Custom Page (via {Flow-Name}.Run method) | Yes | Can be async too | Sync method can wait maximum 120 seconds for response. |
| 3 | Relying on column modification – manually updating rows from OOTB UI | No | Yes | Every row is connected exactly to 1 flow |
| 4 | Relying on column modification – patching selected records from Modern Commanding using PowerFx | No | Yes | Fast and easy to write |
| 5 | Relying on column modification – patching selected records from Modern Commanding using JS code | No | Yes | Harder to write – but overall UI might be much better (rich set of built in UI dialogs and operations in JS) |
| 6 | Business Events triggering – from Modern Commanding using PowerFx and Environment.{ActionName} | No | Yes | DOES NOT WORK – REPORTED TO MICROSOFT (seems to be a bug) |
| 7 | Business Events triggering – from Modern Commanding using JS and calling Custom API with params | No | Yes | Hard to write for Low Coders (but really powerful) |
| 8 | Business Events triggering – from Custom Page using PowerFx and Environment.{ActionName} | No | Yes | Easy to write and kick off |
| 9 | Run flow from BPF step | No | Yes | You can play with this to customize BPF stages and do some background operations. |