**Custom Data provider**

Virtual entities can be when integrating external data with Dataverse, and how we can take that integration even further by adding CRUD support through an embedded canvas app.

The idea would be to get data from SQL, to surface it in Dataverse as a virtual entity, and, also, to allow at least basic search functionality directly in Dataverse over that entity:

Graphical user interface, text, application, email

Description automatically generated

Here is the summary of what needs to be done:

* Create a virtual entity and define all the fields
* Create a custom data provider for that virtual entity
* Configure a virtual entity data source
* Connect this virtual entity to the newly configured data source

**With that done, let’s get back to the Data verse side.**

**1. Setting up the virtual entity**

This part is very straightforward. It seems we still must use classic designer for the virtual entities, but, other than that, you just need to create a new entity and mark it as a virtual entity:

Graphical user interface

Description automatically generated with medium confidence

We are going to use our own custom provider, so I’m not sure it’s important what you put into the “External Name” and “External Collection Name”, but, as you can see above, those fields are mandatory, and I put some values there. Have not used them anywhere else, though.

Leave **“Data Source”** empty for now.

This entity is going to have 5 fields:

Graphical user interface, application

Description automatically generated

Basically, those fields correspond to the SQL columns (they don’t have to – the mapping is done in the custom provider anyway). And the “name” field is there for any entity.

Graphical user interface, text, application

Description automatically generated

All other fields are set up in the same way. Again, when using a custom provider, “External Name” does not have to be populated.

As with any entity, it might be a good idea to configure some views, so I added those columns to the view below:

Graphical user interface, application

Description automatically generated

Finally, don’t forget to add that entity to a model-driven application so you could see your entity somewhere:

Graphical user interface, text, application, Word, email

Description automatically generated

Save, publish all, make sure the application is there and the entity is there.

Just don’t expect it to work yet – we still have not defined the data source. If you try looking at the entity data at this point, you’ll get an ugly error message. Which is fine, just keep moving on.

**2. Setting up the data provider**

We will need a bit more, though, but let me summarize how it works first.

* There is a plugin that works on RetrieveMultiple and on Retrieve
* That plugin is working in stage 30
* We only need to register the plugin – there is no need to register the steps. It’ll happen automatically when registering a data provider
* For most of the registration steps, we’ll be using plugin registration tool

And here are a couple of observations, btw:

* Stage 30 will always get “Query” as a QueryExpression. Even if the original query came in as a Fetch. This is different from Stage 20, where that query would still be in Fetch format. This is why we don’t need to convert one format to another in stage 30
* We’ll need this plugin to support basic search, and, for that, **do not forget to go back to the virtual entity configuration and add some fields to the “Quick Find” view**. You can do it now or later, but make sure you do it

*UPDATED CODE BELOW – IF YOU READ THIS EARLIER, YOU MAY WANT TO REVIEW AGAIN*

With that, here is “Execute” method of my plugin: