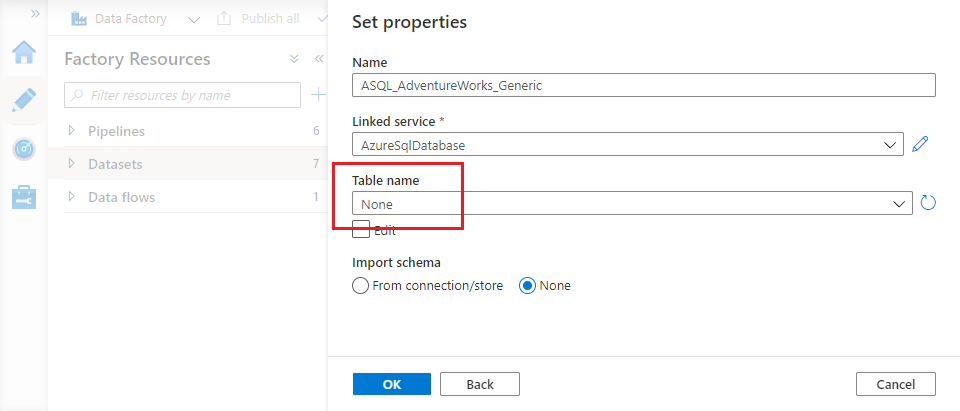
Lab 5 – Make a pipeline generic

Recall that the pipeline you built in lab 2 extracted data from a database table to a file in the data lake. In lab 4.2 you cloned that pipeline to perform the same task for a different table. In this lab you will use dataset parameters and ADF pipeline expressions to create a generic pipeline – a reusable pipeline that can perform the task for *any* table.

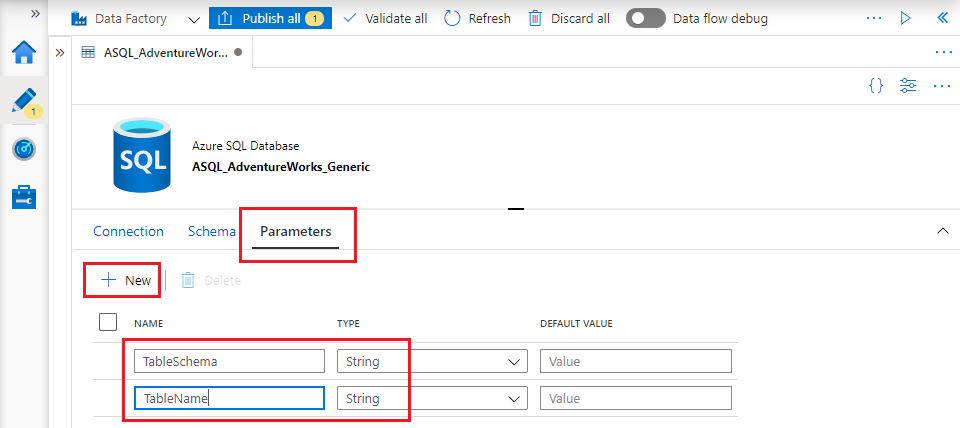
# Lab 5.1 – Create a generic SQL table dataset

Start by creating a new Azure SQL Database dataset, in the same way as in lab 2.2.

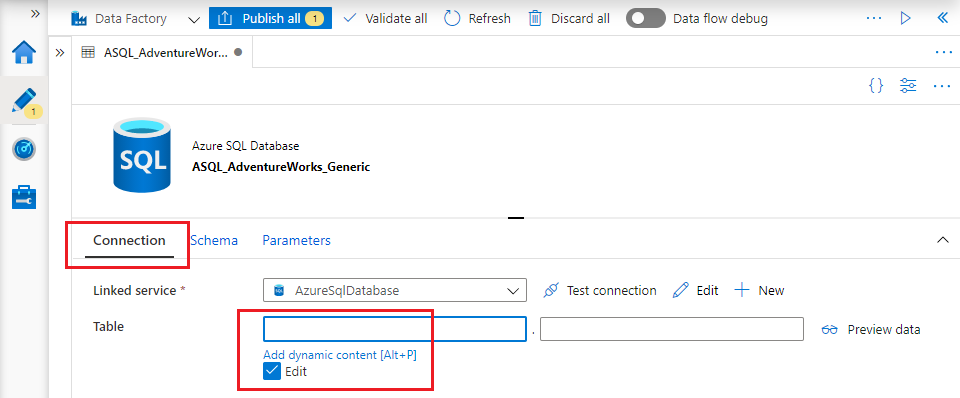
1. In the “Factory resources” sidebar, click the “+” button to the right of “Filter resources by name”, then choose “Dataset”. Select the “Azure SQL Database” data store and click “Continue”.
2. Name the dataset “ASQL\_AdventureWorks\_Generic”, to make the dataset’s purpose clear. Choose your Azure SQL Database from the “Linked service” dropdown. This time however, **do not** choose any table from the “Table name” dropdown – just click “OK”.



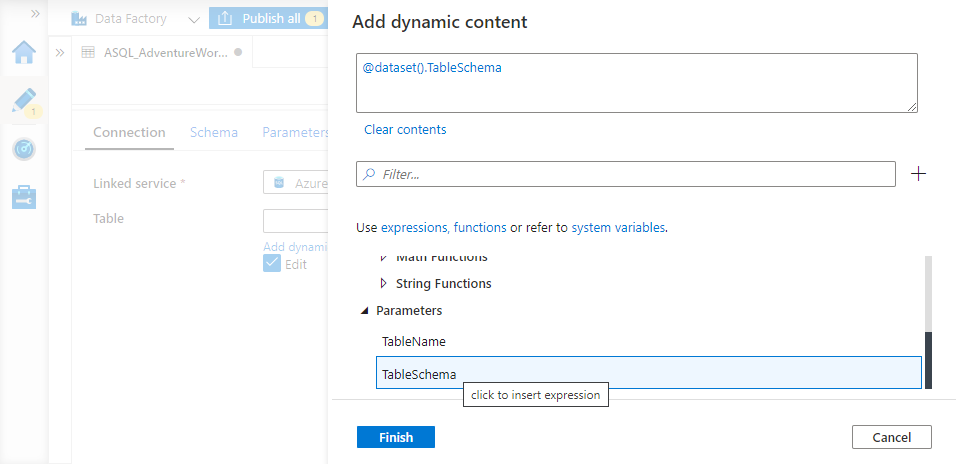
1. The dataset opens in the ADF UX. Close its “Properties” blade using the slider icon in the top right, then select the “Parameters” tab in the configuration pane.
2. Use the “+ New” button to add a new parameter. Name it “TableSchema” and ensure its type is “String”. Add a second string parameter called “TableName”.



1. On the “Connection” tab, under the “Table” dropdown, tick the “Edit” checkbox. The dropdown is replaced by two text fields. Click into the first – a link with the text “Add dynamic content [Alt + P]” appears below the field.



1. Click the link to open the pipeline expression builder. The builder blade appears to the right. Scroll down the list of functions to find the “Parameters” section, then click on “TableSchema” to select the parameter. The expression “@dataset().TableSchema” appears in the expression pane – click “Finish”.



1. Repeat the process for the second field:
   * Click into the field
   * Open the expression builder
   * Select the “TableName” parameter – the expression “@dataset().TableName” appears in the expression pane.
   * Click “Finish”

The dataset you have created contains no hard-coded table name, but instead uses its TableSchema and TableName parameters to specify the parts of a table’s name. Different values can be provided for these parameters at runtime, enabling the same dataset to represent many different tables.

# Lab 5.2 – Create a generic data lake dataset

In the same way as you have created a dataset to represent any AdventureWorks table, now create a generic dataset to represent a data lake file.

1. Create a new dataset for data store “Azure Data Lake Storage Gen2” with file format “DelimitedText”.
2. On the “Set properties” blade, name the dataset “ADLS\_Raw\_Generic” and select your Azure Data Lake Storage linked service. Tick “First row as header”, but leave everything else as is. Click “OK”.
3. Create a parameter for the new dataset of type “String”. Name it “File”.
4. On the “Connection” tab, set the three parts of the “File path” as follows:

* Set “File System” to “lakeroot”
* Set “Directory” to “Raw”
* Use the expression builder to set the “File” field to use the dataset parameter. (Note that if you just type the expression into the field, the ADF UX will interpret the value as a string literal).

1. Save/publish your changes.

