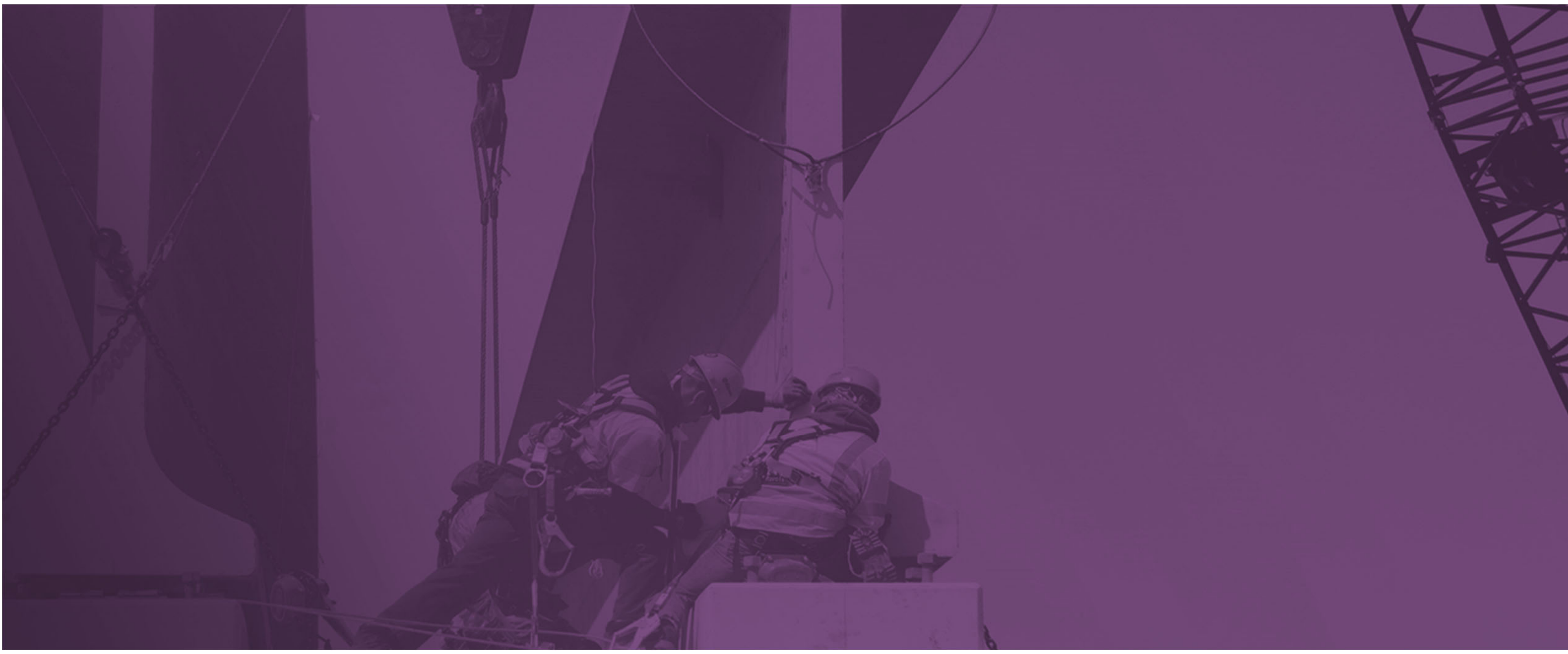


Display Images from InEight Applications in Power BI

User Guide

Changelog

This changelog contains only significant or other notable changes to the document revision. Editorial or minor changes that do not affect the context of the document are not included in the changelog.

Revision	Change Date	Description	Author
0	14/09/2023	Original Draft	Andy Toraman

Contents

Introduction

The purpose of this document is to explain how to get images captured in InEight Applications displayed in Power BI reports.

IMPORTANT: The image URL must already be available in the Project Suite Reporting APIs.

Generating the Access Token

The images from InEight Applications are OAuth protected. This means that viewing these images requires Azure Active Directory (Azure AD) authentication. In this context, OAuth serves as the authorization protocol, and Azure AD acts as the identity provider. When a user attempts to access these images in Power BI / Power Query, they first need to authenticate themselves through Azure AD, proving their identity. Once authenticated, they receive an access token, a temporary credential that grants them permission to view the protected images.

This approach enhances security and control over sensitive image assets, making it suitable for scenarios where access needs to be tightly controlled and authenticated against Azure AD.

Requirements

The following are **required** in order to prove the identity and get the access token. You can contact InEight to get these variables which include the end user's InEight APIM subscription key as well as the Azure tenant service principles.

- APIM SubscriptionKey
- TenantPrefix
- TenantId
- ClientId
- SSOClientId
- ClientSecret

These then will be defined as parametres in Power BI.

How to get the Subscription Key

The subscription key will be acquired from InEight's APIM (Application Programming Interface Management) platform that allows organizations to manage and control the InEight APIs (Application Programming Interfaces) that are used to connect and interact with their software applications.

InEight APIs are gated by the APIM whose key functions include securing APIs through authentication and authorization, routing as well as version control.

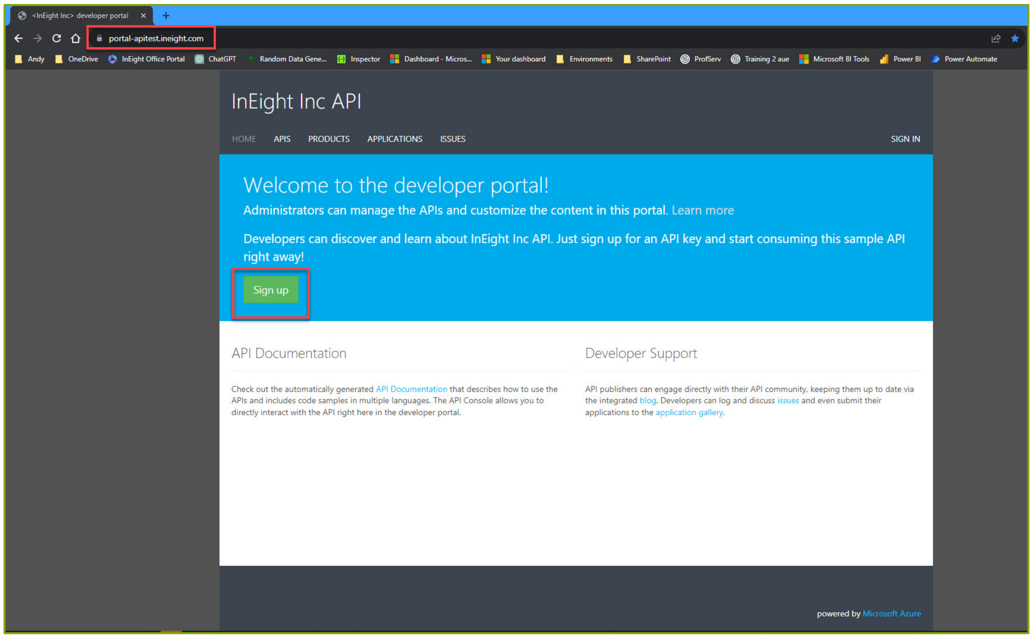
Regarding the Subscription Key that is to be acquired from the APIM, the key consideration is the InEight environment (i.e., the Project Suite or the Cloud Platform) to which the connection will be made with.

In the context of this user guide, if the image data is to come from the **TST environment**, then the subscription key must be acquired from the https://portal-apitest.ineight.com/.

However, if the connection is to be made with the **Production Environment**, then the user needs to register in https://portal.ineight.com/ and acquire the subscription key from there.

The following is an example where the user is willing to connect with the TST environment.

Step 1: Go to https://portal-apitest.ineight.com/ and Sign Up



InEight Inc API

Welcome to the developer portal!

Administrators can manage the APIs and customize the content in this portal. Learn more

Developers can discover and learn about InEight Inc API. Just sign up for an API key and start consuming this sample API right away!

API Documentation

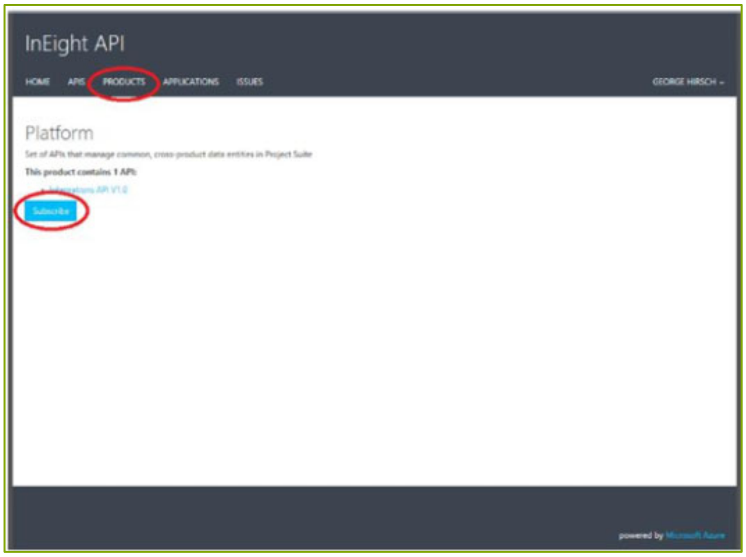
Check out the automatically generated API Documentation that describes how to use the APIs and includes code samples in multiple languages. The API Console allows you to directly interact with the API right here in the developer portal.

Developer Support

API publishers can engage directly with their API community, keeping them up to date via the integrated blog. Developers can log and discuss issues and even submit their applications to the application gallery.

powered by Microsoft Azure

Step 2: Select the Products and then click Subscribe



InEight API

Platform

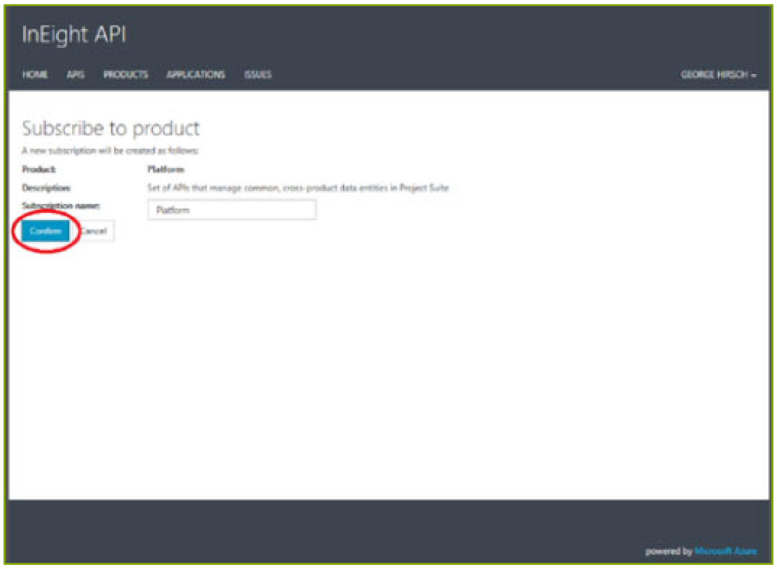
Set of APIs that manage common, cross-product data entities in Project Suite

This product contains 1 API:

- Integrations API V1.0

powered by Microsoft Azure

Step 3: On the Subcribe to Product page, click Confirm

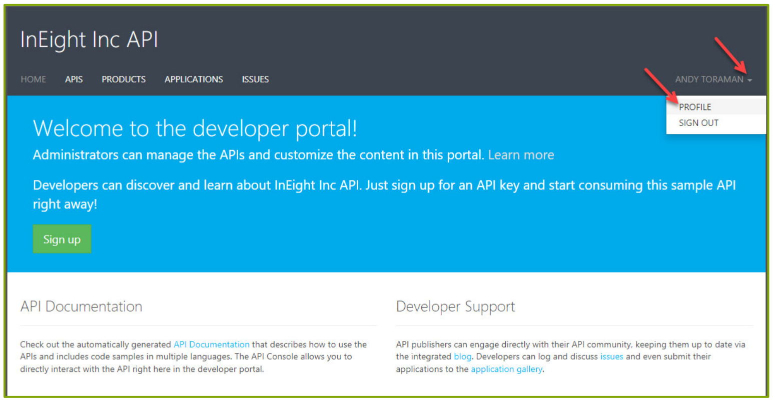

InEight API
HOME APIS PRODUCTS APPLICATIONS ISSUES
GEORGE HIRSCH
Confirm
Cancel
powered by Microsoft Azure

Subscribe to product

A new subscription will be created as follows:

Product:	Platform
Description:	Set of APIs that manage common, cross-product data entities in Project Suite
Subscription name:	Platform

Step 4: Select the Dropdown where your name is displayed and click PROFILE



Welcome to the developer portal!

Administrators can manage the APIs and customize the content in this portal. Learn more

Developers can discover and learn about InEight Inc API. Just sign up for an API key and start consuming this sample API right away!

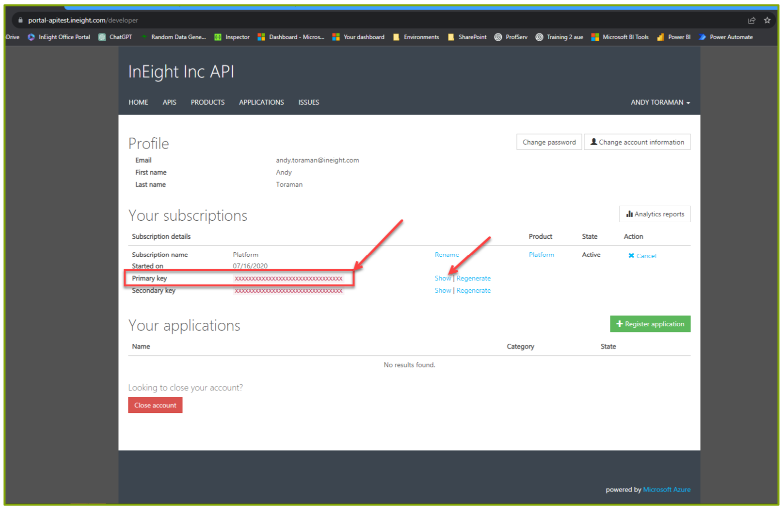
API Documentation

Check out the automatically generated API Documentation that describes how to use the APIs and includes code samples in multiple languages. The API Console allows you to directly interact with the API right here in the developer portal.

Developer Support

API publishers can engage directly with their API community, keeping them up to date via the integrated blog. Developers can log and discuss issues and even submit their applications to the application gallery.

Step 5: You will copy the Primary Key (i.e. Subscription Key) by clicking Show.


portal-apitest.ineight.com/developer
Drive InEight Office Portal ChatGPT Random Data Gene... Inspector Dashboard - Micros... Your dashboard Environments SharePoint ProfServ Training 2 aue Microsoft BI Tools Power BI Power Automate
InEight Inc API
HOME APIS PRODUCTS APPLICATIONS ISSUES
ANDY TORAMAN
Change password
Change account information
Analytics reports
Register application
Close account
powered by Microsoft Azure

Profile

Email	andy.toraman@ineight.com
First name	Andy
Last name	Toraman

Your subscriptions

Subscription details			Product	State	Action
Subscription name	Platform	Rename	Platform	Active	✖ Cancel
Started on	07/16/2020				
Primary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show | Regenerate			
Secondary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show | Regenerate			

Your applications

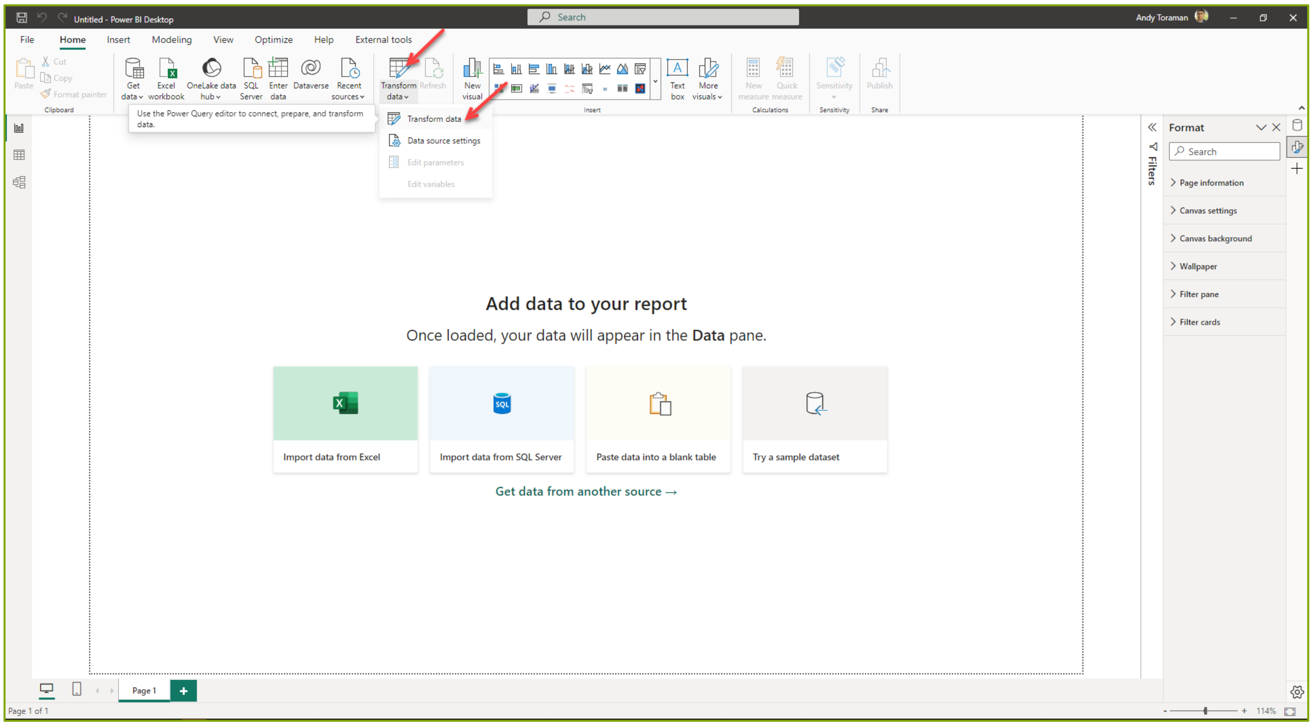
Name	Category	State
No results found.		

Looking to close your account?

Getting the Image Displayed in Power BI

Power Query is a tool whose primary purpose is to streamline the process of importing, transforming, and cleaning data from various sources into a format suitable for analysis or reporting in tools like Excel, Power BI, or SQL Server. The following explains step-by-step process to use Power Query within Power BI desktop tool to generate the access token.

Step 1: Open Power BI Desktop and click Transform Data

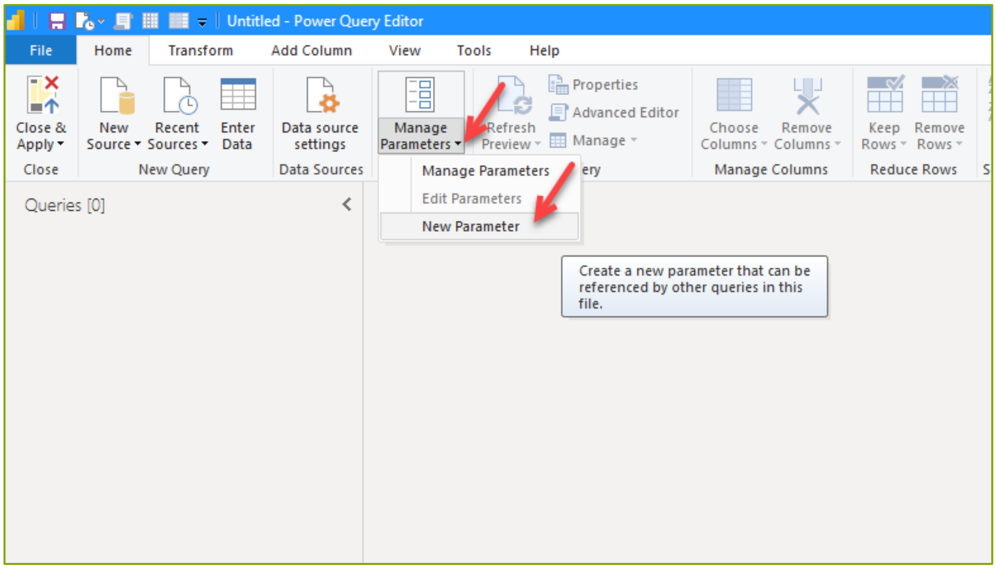


Use the Power Query editor to connect, prepare, and transform data.

Add data to your report

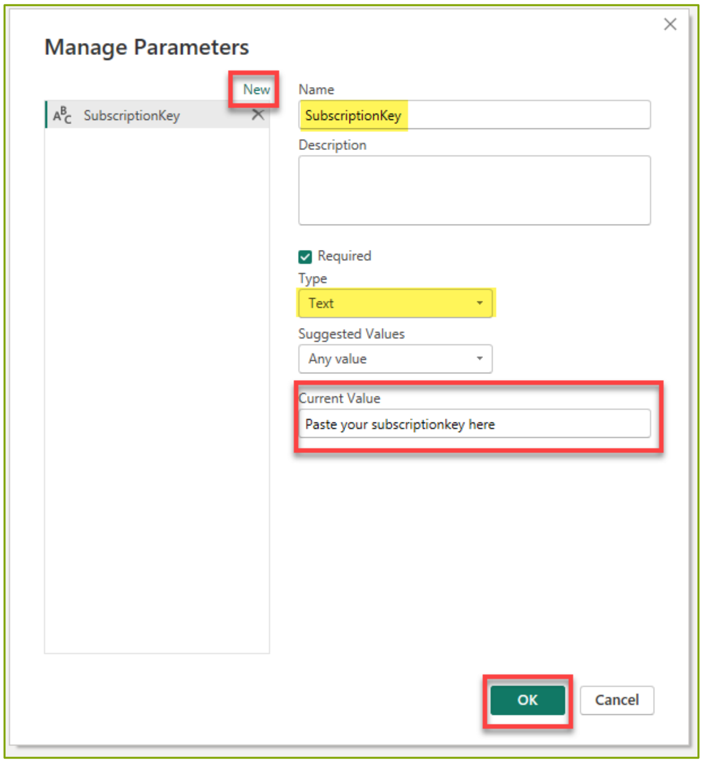
Once loaded, your data will appear in the **Data** pane.

Step 2: Add New Parameter



Create a new parameter that can be referenced by other queries in this file.

Step 3: Add all the variables listed on Page 4 of this document



Manage Parameters

New

SubscriptionKey

Name: SubscriptionKey

Description:

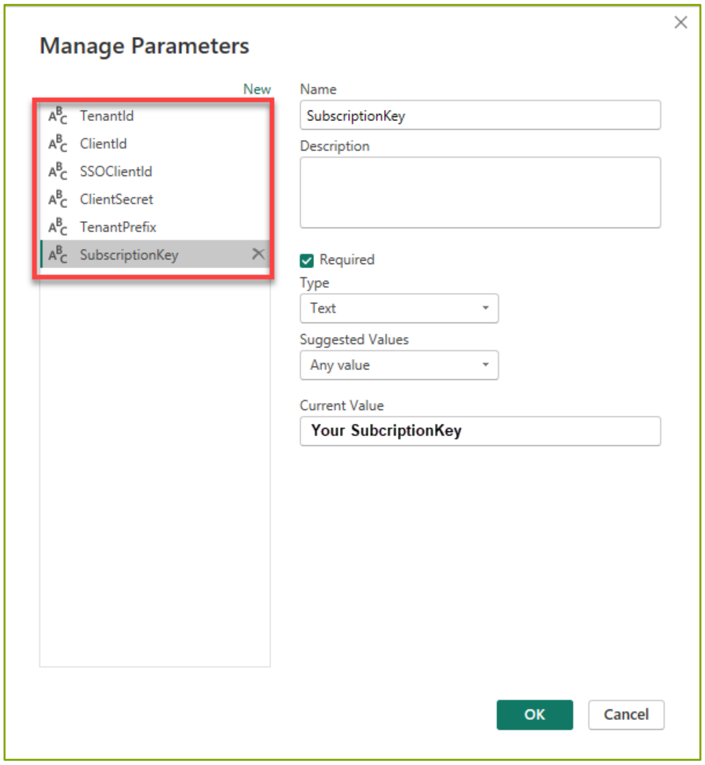
☑ Required

Type: Text

Suggested Values: Any value

Current Value: Paste your subscriptionkey here

Step 4: Make sure your list Parametres look exactly like below



Manage Parameters

New

- TenantId
- ClientId
- SSOClientId
- ClientSecret
- TenantPrefix
- SubscriptionKey

Name: SubscriptionKey

Description:

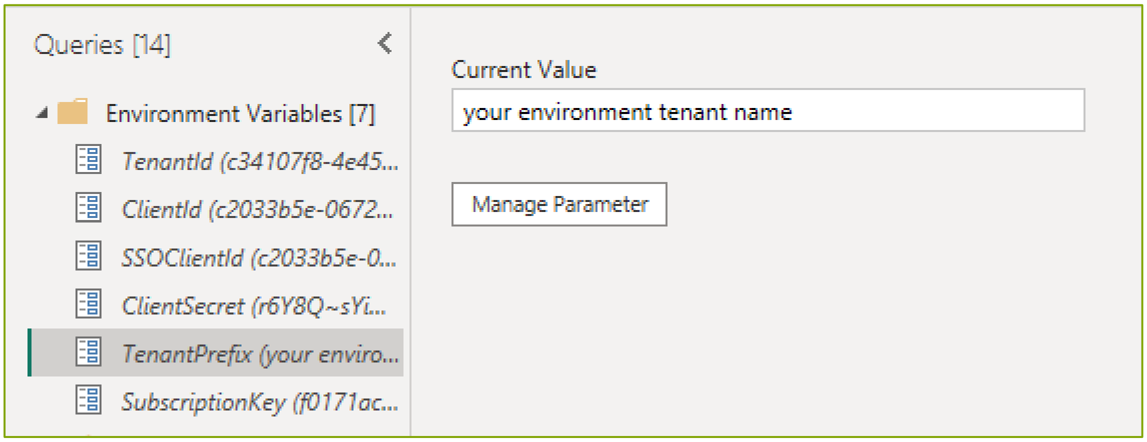
☑ Required

Type: Text

Suggested Values: Any value

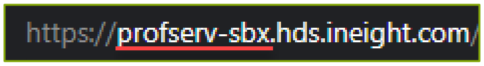
Current Value: **Your SubcriptionKey**

Step 5: The parametres should look like below.


Queries [14]
Environment Variables [7]
TenantId (c34107f8-4e45...
ClientId (c2033b5e-0672...
SSOClientId (c2033b5e-0...
ClientSecret (r6Y8Q~sYi...
TenantPrefix (your enviro...
SubscriptionKey (f0171ac...
Manage Parameter

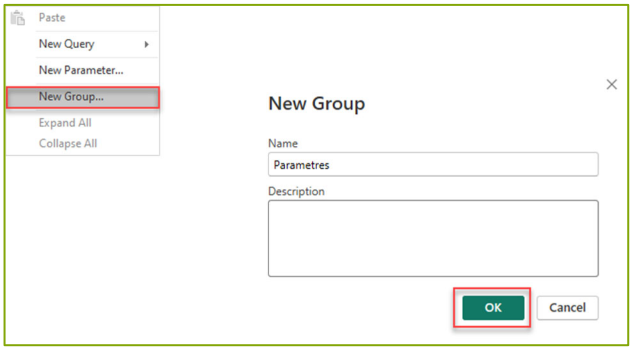
Current Value: your environment tenant name

Your tenant name can easily be acquired from your InEight URL. The screenshot below is an example of which part of the URL corresponds with the tenant name.



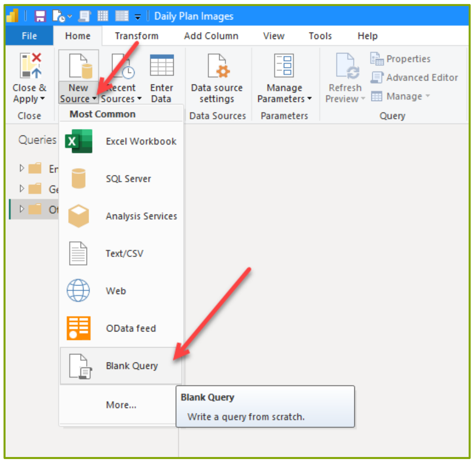
https://profserv-sbx.hds.ineight.com/

OPTIONAL: You can create a group **by right clicking under the queries pane** to get all the parameters within a single folder. This will make it easier to maintain the queries and parameters going forward.

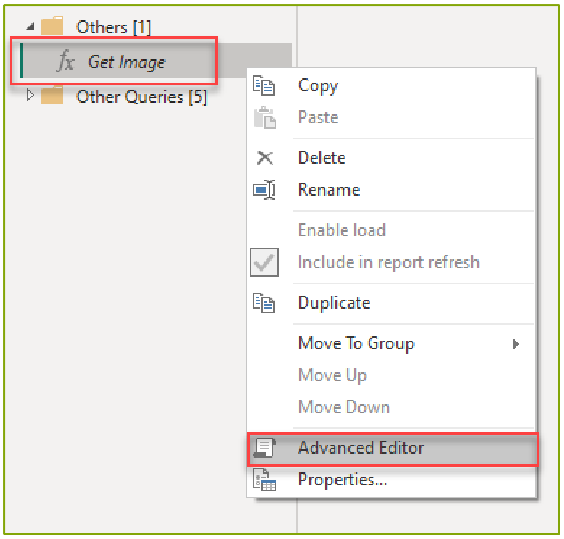


Name: Parametres

Step 6: Create a Blank Query by right clicking under the Queries Pane or using the New Source Dropdown



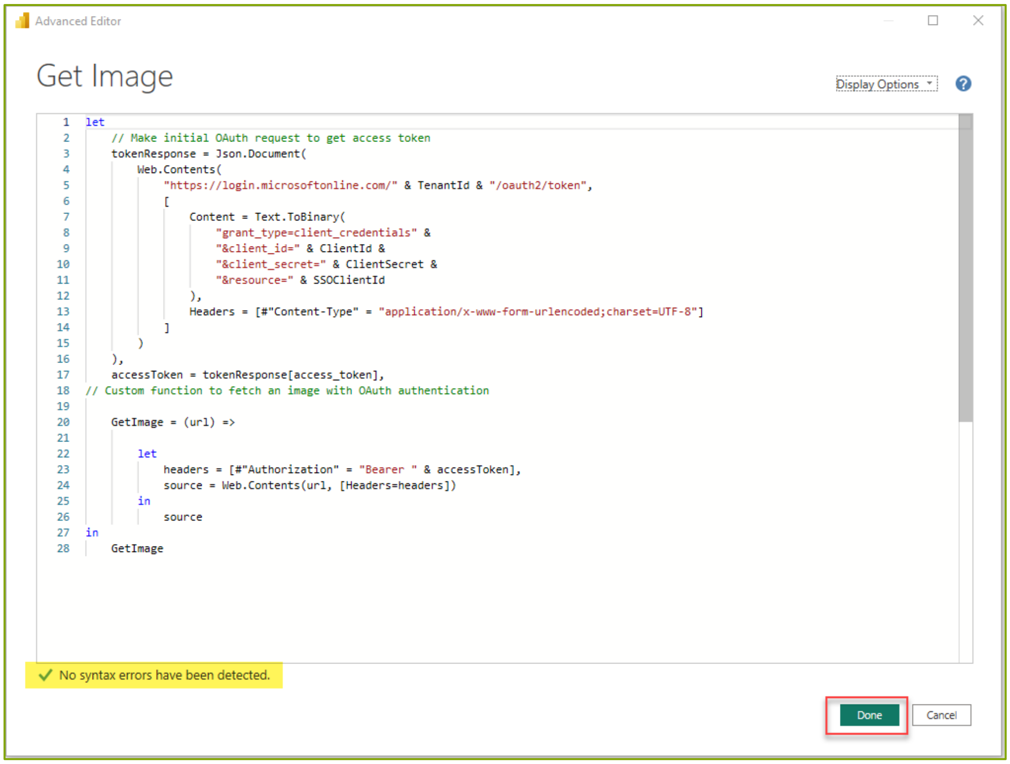
Step 7: Re- name Query1 as Get Image and then open Advanced Editor by Right clicking



Step 8: Delete the default lines and replace it with the following script

```
let
   // Make initial OAuth request to get access token
   tokenResponse = Json.Document(
      Web.Contents(
         "https://login.microsoftonline.com/" & TenantId & "/oauth2/token",
         [
            Content = Text.ToBinary(
               "grant_type=client_credentials" &
               "&client_id=" & ClientId &
               "&client_secret=" & ClientSecret &
               "&resource=" & SSOClientId
            ),
            Headers = [#"Content-Type" = "application/x-www-form-urlencoded;charset=UTF-8"]
         ]
      )
   ),
   accessToken = tokenResponse[access_token],
// Custom function to fetch an image with OAuth authentication
      GetImage = (url) =>

      let
         headers = [#"Authorization" = "Bearer " & accessToken],
         source = Web.Contents(url, [Headers=headers])
      in
         source
in
   GetImage
```

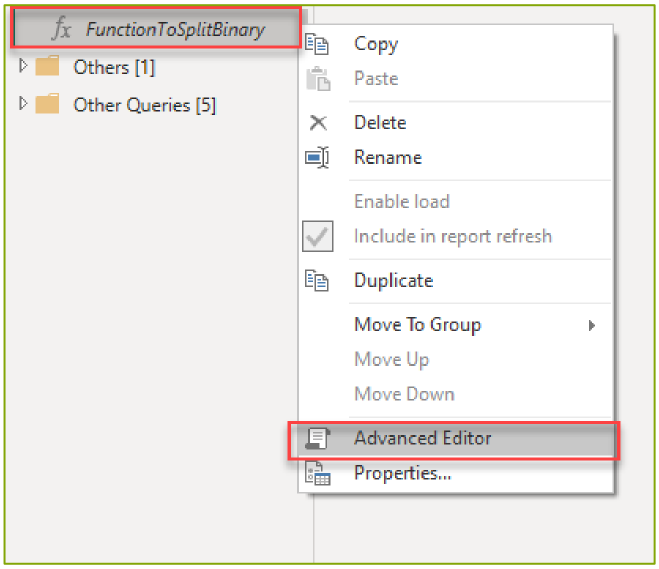
Step 9: Click Done



```
let
    // Make initial OAuth request to get access token
    tokenResponse = Json.Document(
        Web.Contents(
            "https://login.microsoftonline.com/" & TenantId & "/oauth2/token",
            [
                Content = Text.ToBinary(
                    "grant_type=client_credentials" &
                    "&client_id=" & ClientId &
                    "&client_secret=" & ClientSecret &
                    "&resource=" & SSOClientId
                ),
                Headers = [#"Content-Type" = "application/x-www-form-urlencoded;charset=UTF-8"]
            ]
        )
    ),
    accessToken = tokenResponse[access_token],
// Custom function to fetch an image with OAuth authentication

    GetImage = (url) =>

        let
            headers = [#"Authorization" = "Bearer " & accessToken],
            source = Web.Contents(url, [Headers=headers])
        in
            source
in
    GetImage
```

Step 10: Add another query as in Step 6 and re-name it as FunctionToSplitBinary

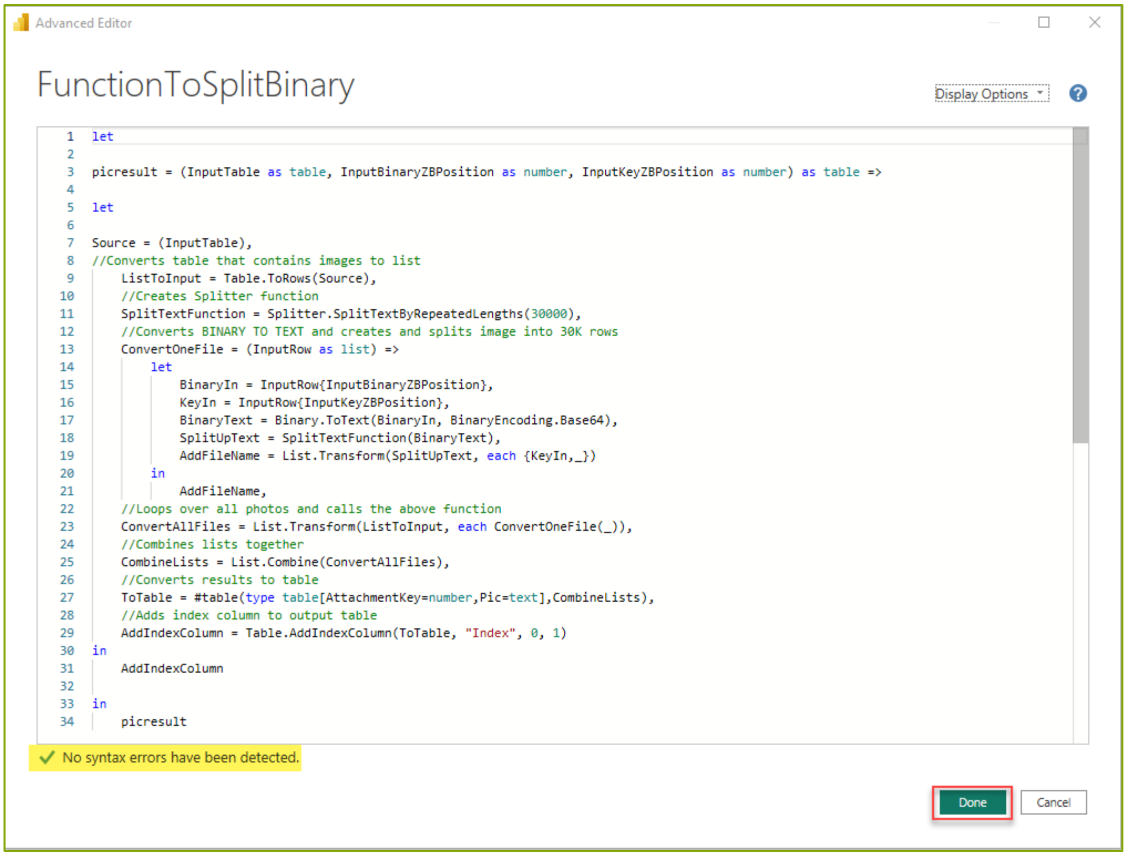
Step 11: Open the Advanced Editor



Step 12: Copy and paste the following script
(credit to Guy in a Cube https://www.youtube.com/watch?v=Q82yzcfkqAc)

```
let
picresult = (InputTable as table, InputBinaryZBPosition as number, InputKeyZBPosition as number) as table =>
let
Source = (InputTable),
//Converts table that contains images to list
       ListToInput = Table.ToRows(Source),
       //Creates Splitter function
  SplitTextFunction = Splitter.SplitTextByRepeatedLengths(30000),
  //Converts BINARY TO TEXT and creates and splits image into 30K rows
  ConvertOneFile = (InputRow as list) =>
    let
      BinaryIn = InputRow{InputBinaryZBPosition},
      KeyIn = InputRow{InputKeyZBPosition},
                BinaryText = Binary.ToText(BinaryIn, BinaryEncoding.Base64),
      SplitUpText = SplitTextFunction(BinaryText),
      AddFileName = List.Transform(SplitUpText, each {KeyIn,_})
    in
      AddFileName,
  //Loops over all photos and calls the above function
  ConvertAllFiles = List.Transform(ListToInput, each ConvertOneFile(_)),
  //Combines lists together
  CombineLists = List.Combine(ConvertAllFiles),
  //Converts results to table
  ToTable = #table(type table[AttachmentKey=number,Pic=text],CombineLists),
  //Adds index column to output table
  AddIndexColumn = Table.AddIndexColumn(ToTable, "Index", 0, 1)
in
  AddIndexColumn
in
       picresult
```

Step 13: Make sure the query looks like below and click Done.

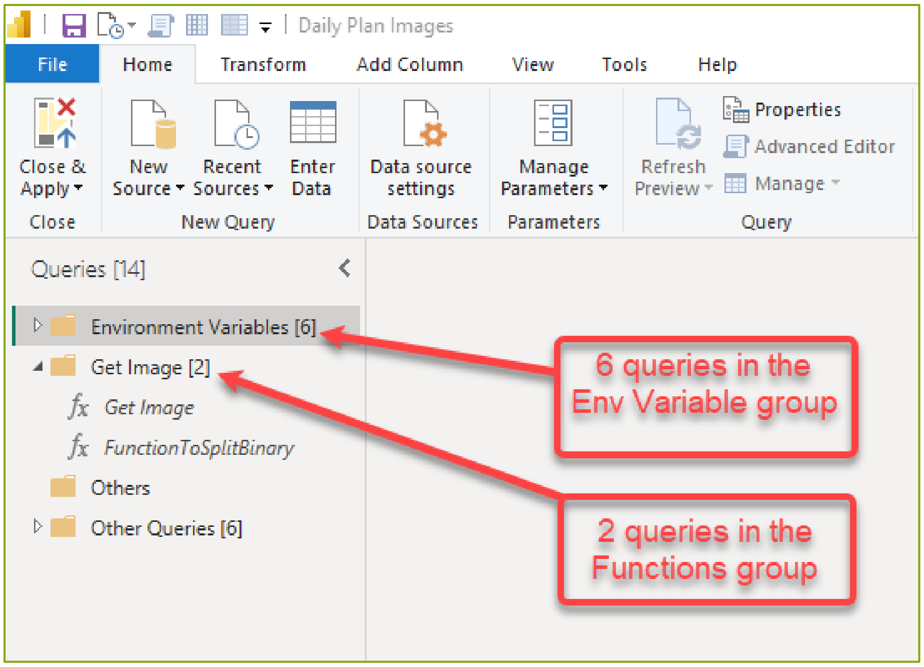


FunctionToSplitBinary

```
let

picresult = (InputTable as table, InputBinaryZBPosition as number, InputKeyZBPosition as number) as table =>

let

Source = (InputTable),
//Converts table that contains images to list
    ListToInput = Table.ToRows(Source),
    //Creates Splitter function
    SplitTextFunction = Splitter.SplitTextByRepeatedLengths(30000),
    //Converts BINARY TO TEXT and creates and splits image into 30K rows
    ConvertOneFile = (InputRow as list) =>
        let
            BinaryIn = InputRow{InputBinaryZBPosition},
            KeyIn = InputRow{InputKeyZBPosition},
            BinaryText = Binary.ToText(BinaryIn, BinaryEncoding.Base64),
            SplitUpText = SplitTextFunction(BinaryText),
            AddFileName = List.Transform(SplitUpText, each {KeyIn,_})
        in
            AddFileName,
    //Loops over all photos and calls the above function
    ConvertAllFiles = List.Transform(ListToInput, each ConvertOneFile(_)),
    //Combines lists together
    CombineLists = List.Combine(ConvertAllFiles),
    //Converts results to table
    ToTable = #table(type table[AttachmentKey=number,Pic=text],CombineLists),
    //Adds index column to output table
    AddIndexColumn = Table.AddIndexColumn(ToTable, "Index", 0, 1)
in
    AddIndexColumn

in
    picresult
```

✓ No syntax errors have been detected.

Step 14: Create a group for the Get Image and FunctionToSplitBinary functions (OPTIONAL)

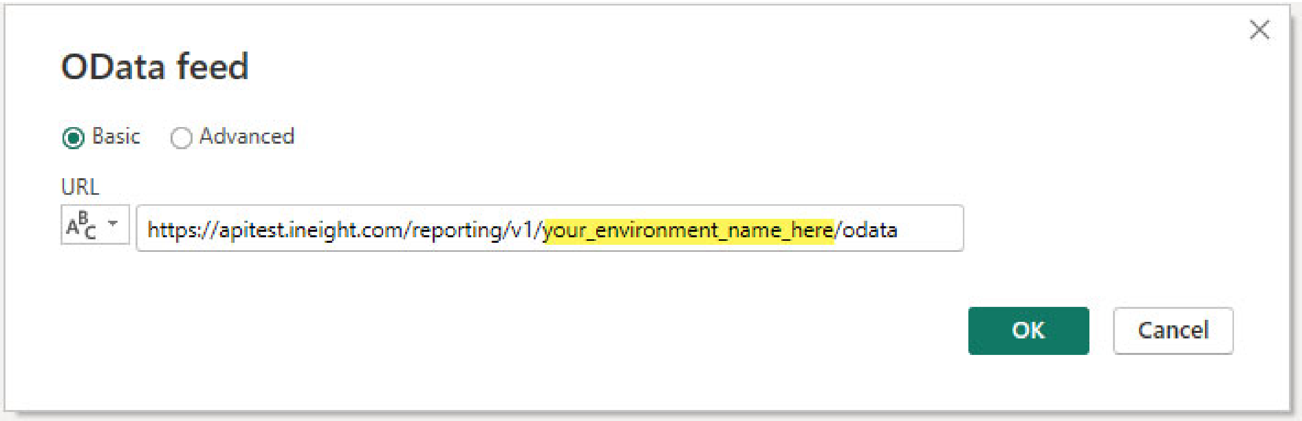


6 queries in the Env Variable group

2 queries in the Functions group

Step 15: Click on New Source and Select OData Feed

Step 16: Enter the URL that allows you to access the Reporting Tables (i.e. Reporting APIs)



OData feed

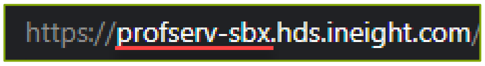
● Basic ○ Advanced

URL

https://apitest.ineight.com/reporting/v1/your_environment_name_here/odata

Make sure you type the name of your InEight environment. In the example below the URL would be the following:

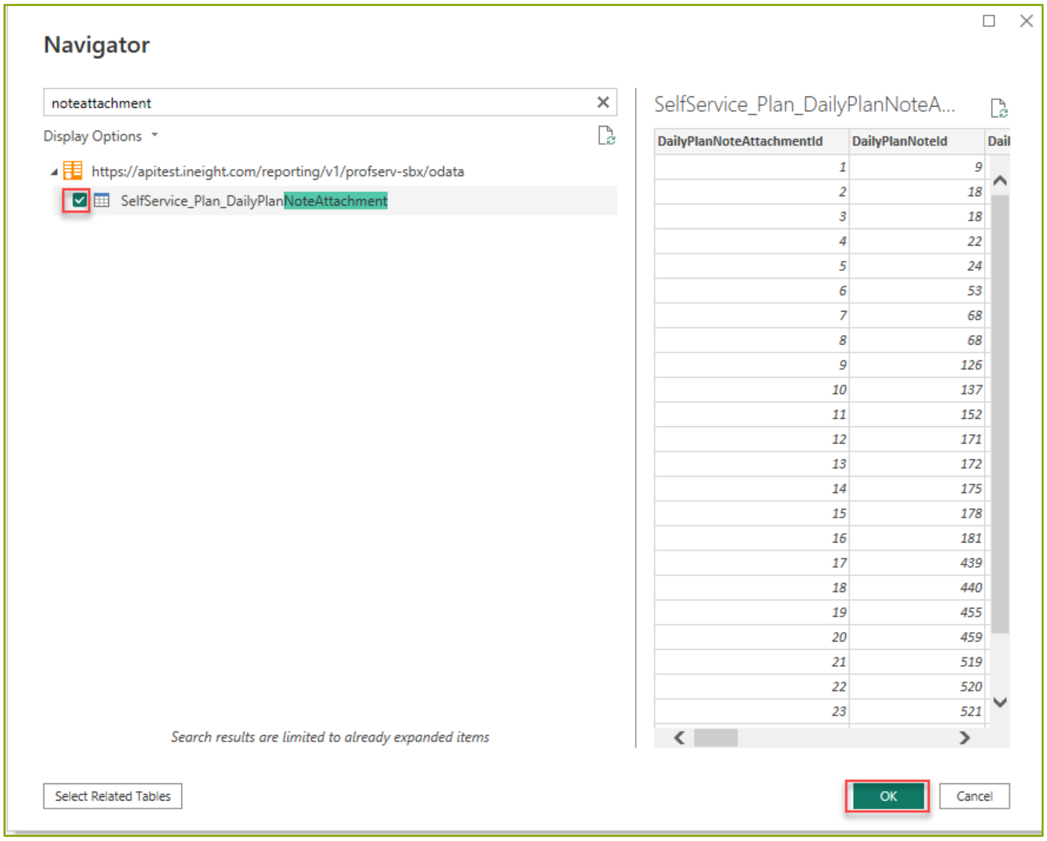
InEight Environment:



https://profserv-sbx.hds.ineight.com/

OData URL: *https://apitest.ineight.com/reporting/v1/profserv-sbx/odata*

Step 17: *Type NoteAttachment in the search box of the Navigator window and Select and Click OK*



Navigator

noteattachment

Display Options

https://apitest.ineight.com/reporting/v1/profserv-sbx/odata

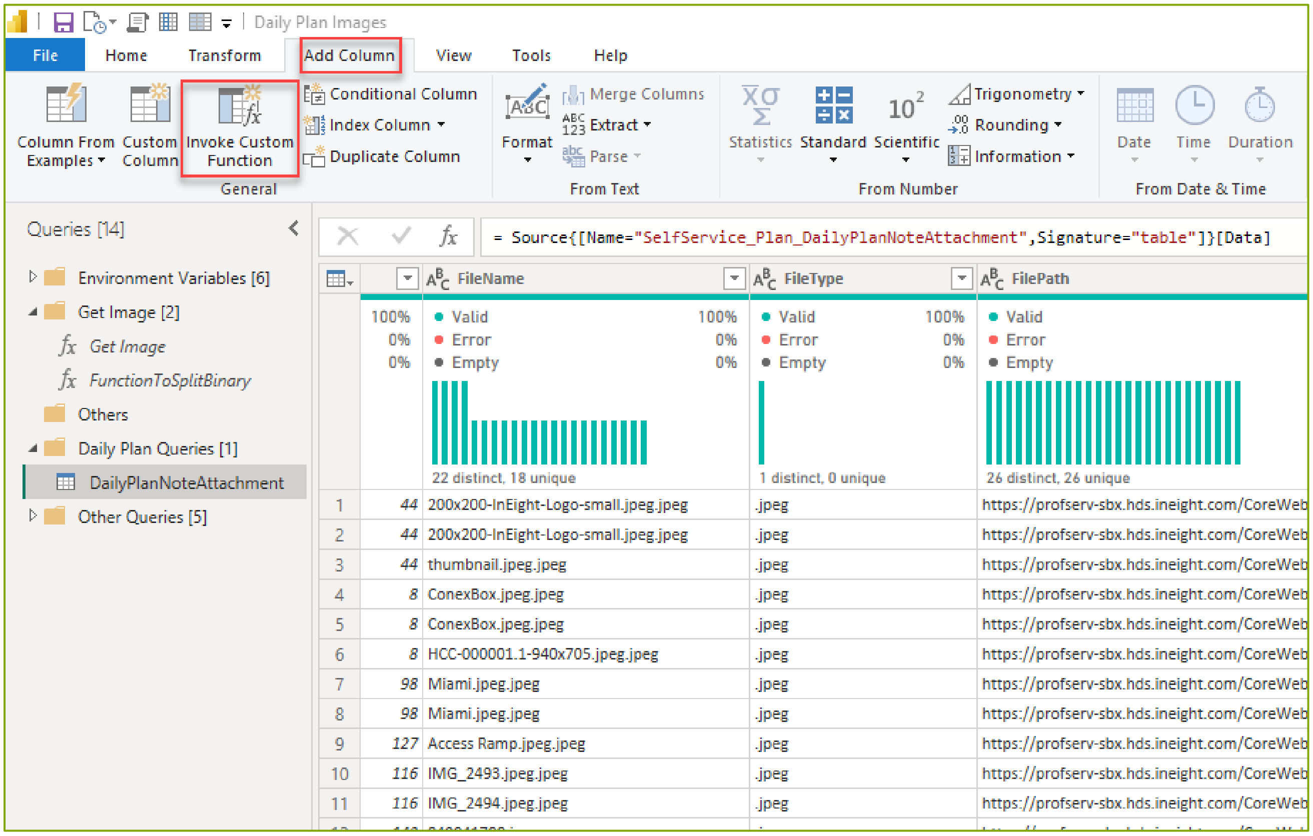
☑ SelfService_Plan_DailyPlanNoteAttachment

Search results are limited to already expanded items

SelfService_Plan_DailyPlanNoteA...

DailyPlanNoteAttachmentId	DailyPlanNoteId	Dail
1	9	
2	18	
3	18	
4	22	
5	24	
6	53	
7	68	
8	68	
9	126	
10	137	
11	152	
12	171	
13	172	
14	175	
15	178	
16	181	
17	439	
18	440	
19	455	
20	459	
21	519	
22	520	
23	521	

Step 18: Select “Add Column” and click “Invoke Custom Function”



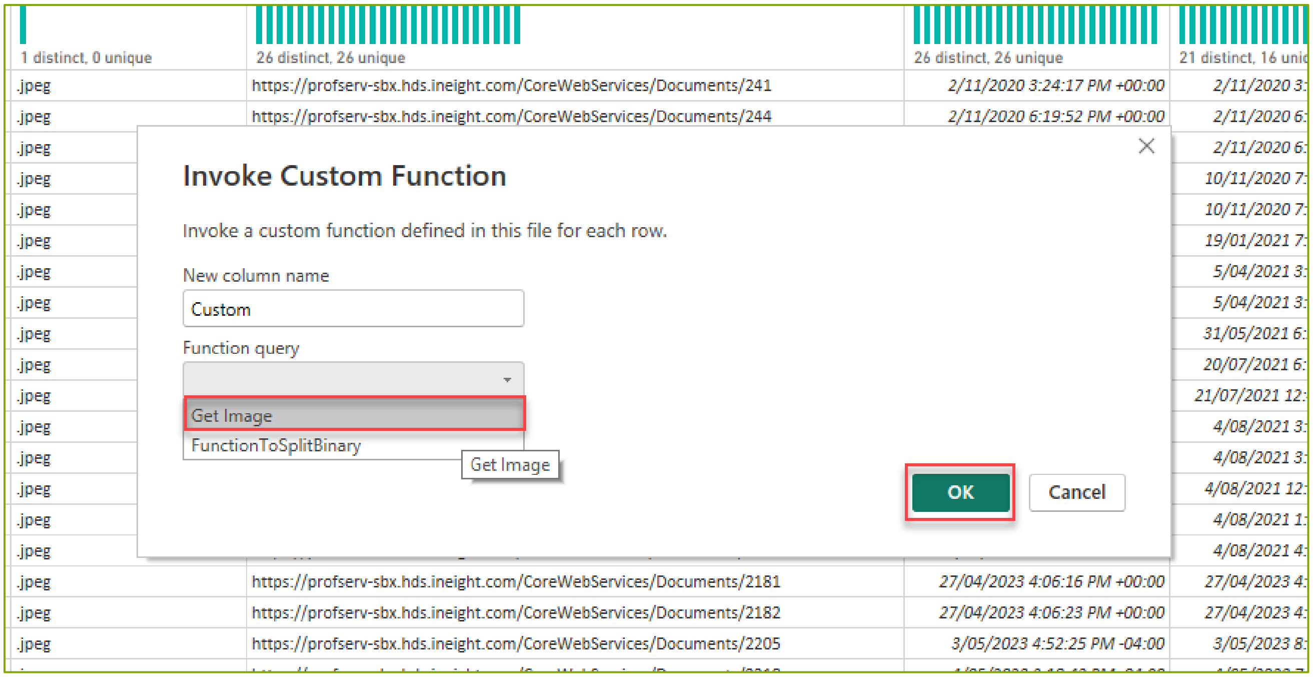
Queries [14]

- Environment Variables [6]
- Get Image [2]
  - Get Image
  - FunctionToSplitBinary
- Others
- Daily Plan Queries [1]
  - DailyPlanNoteAttachment
- Other Queries [5]

```
= Source{[Name="SelfService_Plan_DailyPlanNoteAttachment",Signature="table"]}[Data]
```

		FileName	FileType	FilePath
		100% Valid 0% Error 0% Empty	100% Valid 0% Error 0% Empty	100% Valid 0% Error 0% Empty
		22 distinct, 18 unique	1 distinct, 0 unique	26 distinct, 26 unique
1	44	200x200-InEight-Logo-small.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
2	44	200x200-InEight-Logo-small.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
3	44	thumbnail.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
4	8	ConexBox.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
5	8	ConexBox.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
6	8	HCC-000001.1-940x705.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
7	98	Miami.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
8	98	Miami.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
9	127	Access Ramp.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
10	116	IMG_2493.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb
11	116	IMG_2494.jpeg.jpeg	.jpeg	https://profserv-sbx.hds.ineight.com/CoreWeb

Step 19: Select Get Image function



1 distinct, 0 unique	26 distinct, 26 unique	26 distinct, 26 unique	21 distinct, 16 uniq
.jpeg	https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/241	2/11/2020 3:24:17 PM +00:00	2/11/2020 3:
.jpeg	https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/244	2/11/2020 6:19:52 PM +00:00	2/11/2020 6:
.jpeg			2/11/2020 6:
.jpeg			10/11/2020 7:
.jpeg			10/11/2020 7:
.jpeg			19/01/2021 7:
.jpeg			5/04/2021 3:
.jpeg			5/04/2021 3:
.jpeg			31/05/2021 6:
.jpeg			20/07/2021 6:
.jpeg			21/07/2021 12:
.jpeg			4/08/2021 3:
.jpeg			4/08/2021 3:
.jpeg			4/08/2021 12:
.jpeg			4/08/2021 1:
.jpeg			4/08/2021 4:
.jpeg	https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/2181	27/04/2023 4:06:16 PM +00:00	27/04/2023 4:
.jpeg	https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/2182	27/04/2023 4:06:23 PM +00:00	27/04/2023 4:
.jpeg	https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/2205	3/05/2023 4:52:25 PM -04:00	3/05/2023 8:

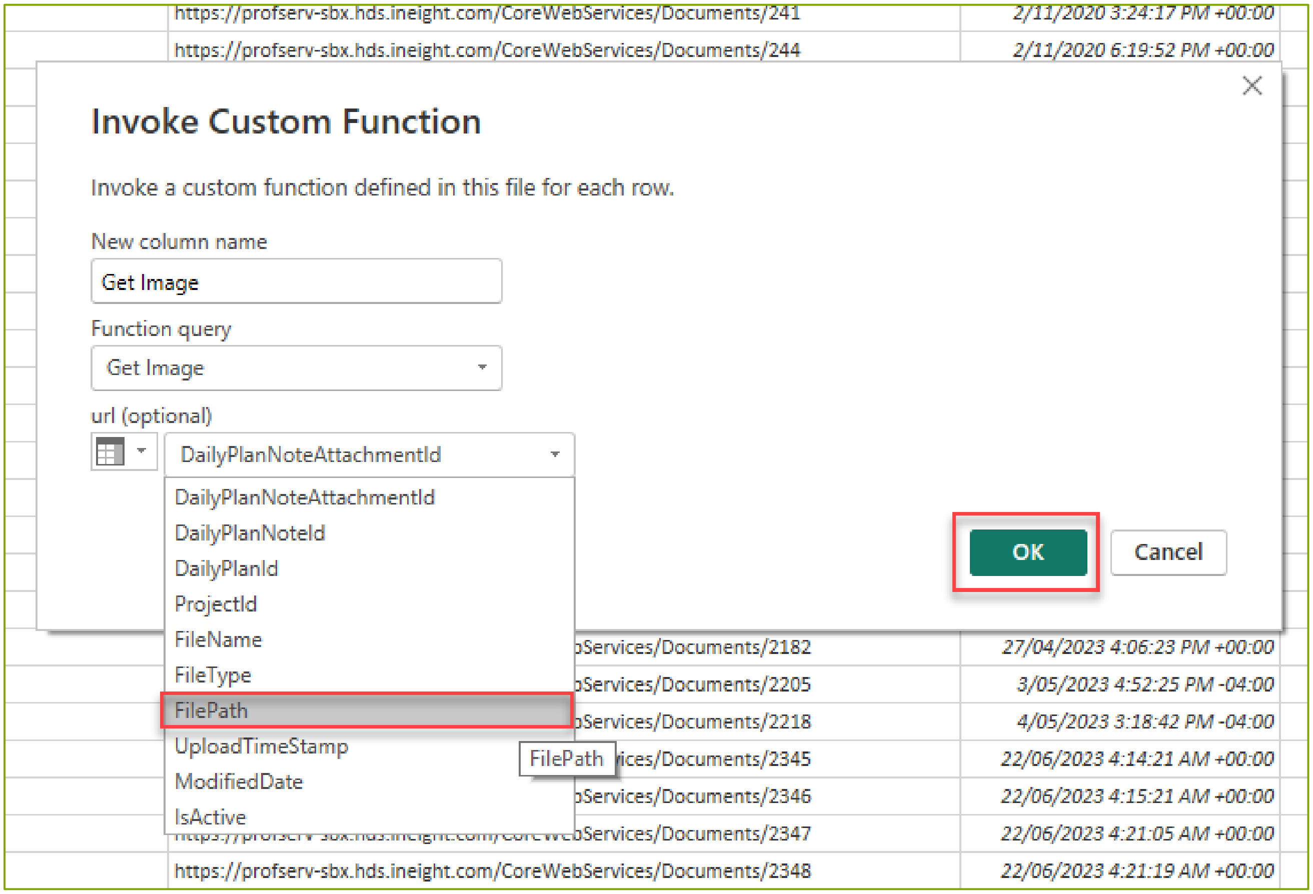
Invoke Custom Function

Invoke a custom function defined in this file for each row.

New column name: Custom

Function query: Get Image

Step 20: Select FilePath for the url dropdown and click OK



https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/241	2/11/2020 3:24:17 PM +00:00
https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/244	2/11/2020 6:19:52 PM +00:00
bServices/Documents/2182	27/04/2023 4:06:23 PM +00:00
bServices/Documents/2205	3/05/2023 4:52:25 PM -04:00
bServices/Documents/2218	4/05/2023 3:18:42 PM -04:00
ices/Documents/2345	22/06/2023 4:14:21 AM +00:00
bServices/Documents/2346	22/06/2023 4:15:21 AM +00:00
bServices/Documents/2347	22/06/2023 4:21:05 AM +00:00
https://profserv-sbx.hds.ineight.com/CoreWebServices/Documents/2348	22/06/2023 4:21:19 AM +00:00

Invoke Custom Function

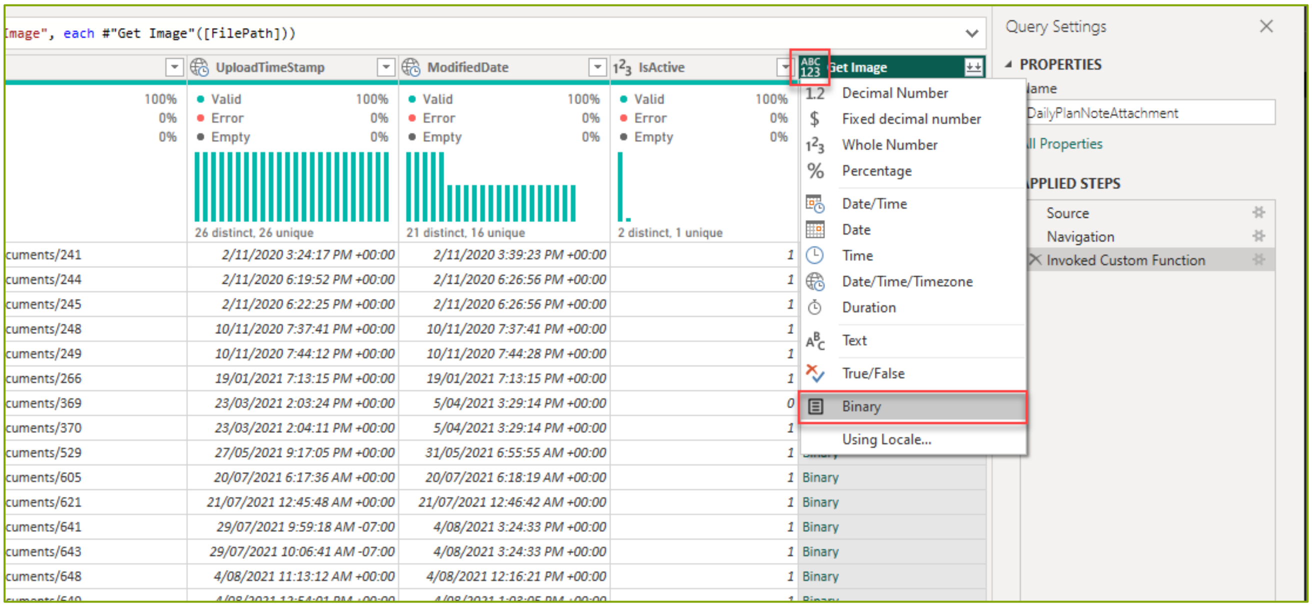
Invoke a custom function defined in this file for each row.

New column name: Get Image

Function query: Get Image

url (optional): DailyPlanNoteAttachmentId

Step 21: Changed the Data Type of the Get Image column to Binary by clicking on ABC/123 in the header



```
Image", each #"Get Image"([FilePath]))
```

	UploadTimeStamp	ModifiedDate	IsActive	Get Image
100%	Valid 100%	Valid 100%	Valid 100%	100%
0%	Error 0%	Error 0%	Error 0%	0%
0%	Empty 0%	Empty 0%	Empty 0%	0%
	26 distinct, 26 unique	21 distinct, 16 unique	2 distinct, 1 unique	
cuments/241	2/11/2020 3:24:17 PM +00:00	2/11/2020 3:39:23 PM +00:00	1	
cuments/244	2/11/2020 6:19:52 PM +00:00	2/11/2020 6:26:56 PM +00:00	1	
cuments/245	2/11/2020 6:22:25 PM +00:00	2/11/2020 6:26:56 PM +00:00	1	
cuments/248	10/11/2020 7:37:41 PM +00:00	10/11/2020 7:37:41 PM +00:00	1	
cuments/249	10/11/2020 7:44:12 PM +00:00	10/11/2020 7:44:28 PM +00:00	1	
cuments/266	19/01/2021 7:13:15 PM +00:00	19/01/2021 7:13:15 PM +00:00	1	
cuments/369	23/03/2021 2:03:24 PM +00:00	5/04/2021 3:29:14 PM +00:00	0	
cuments/370	23/03/2021 2:04:11 PM +00:00	5/04/2021 3:29:14 PM +00:00	1	
cuments/529	27/05/2021 9:17:05 PM +00:00	31/05/2021 6:55:55 AM +00:00	1	
cuments/605	20/07/2021 6:17:36 AM +00:00	20/07/2021 6:18:19 AM +00:00	1	Binary
cuments/621	21/07/2021 12:45:48 AM +00:00	21/07/2021 12:46:42 AM +00:00	1	Binary
cuments/641	29/07/2021 9:59:18 AM -07:00	4/08/2021 3:24:33 PM +00:00	1	Binary
cuments/643	29/07/2021 10:06:41 AM -07:00	4/08/2021 3:24:33 PM +00:00	1	Binary
cuments/648	4/08/2021 11:13:12 AM +00:00	4/08/2021 12:16:21 PM +00:00	1	Binary

Query Settings

PROPERTIES

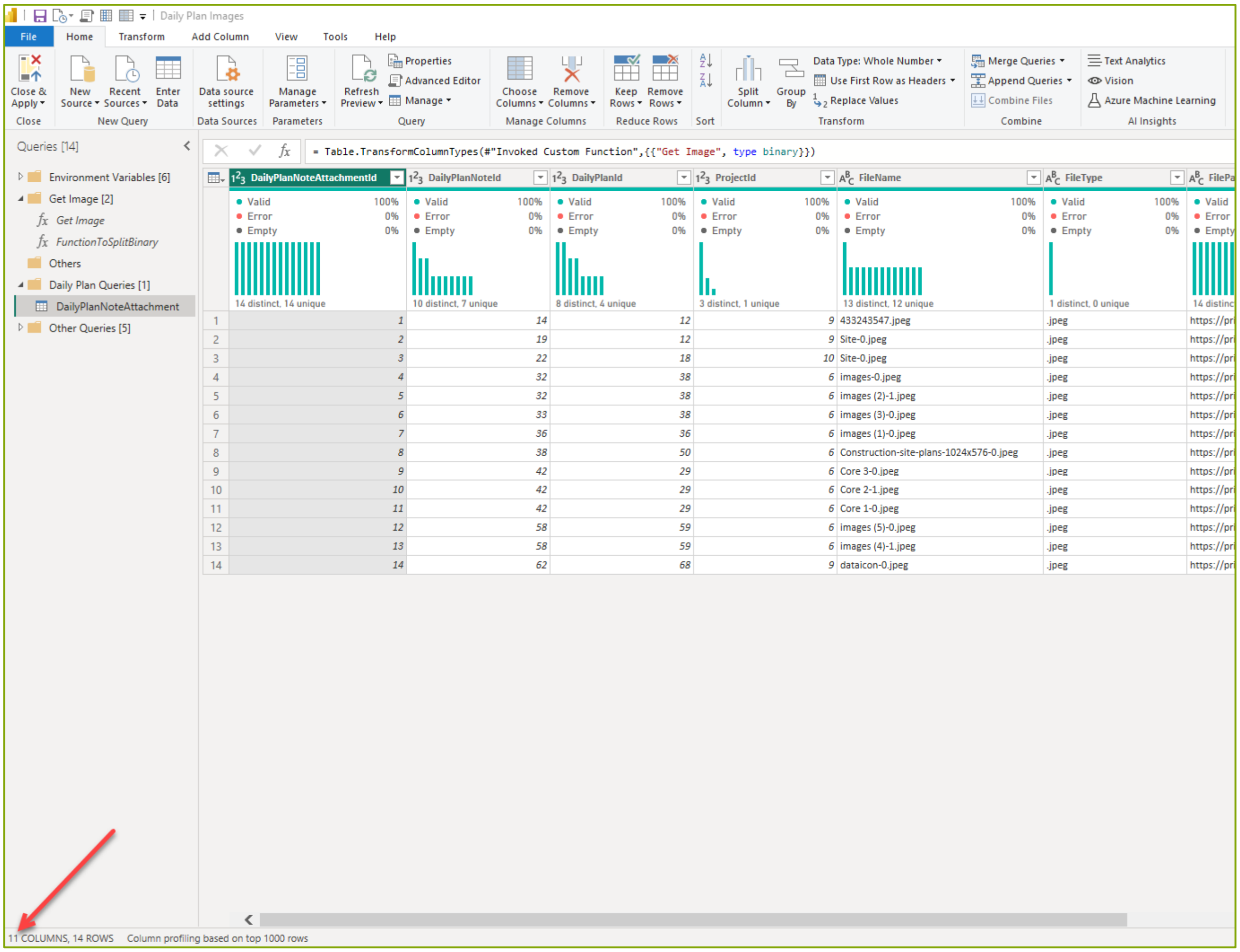
Name: DailyPlanNoteAttachment

All Properties

APPLIED STEPS

- Source
- Navigation
- Invoked Custom Function

Step 22: Check the # of total columns in the table from bottom left corner



```
= Table.TransformColumnTypes(#"Invoked Custom Function",{{"Get Image", type binary}})
```

	DailyPlanNoteAttachmentId	DailyPlanNoteId	DailyPlanId	ProjectId	FileName	FileType	FilePa
	● Valid 100% ● Error 0% ● Empty 0% 14 distinct, 14 unique	● Valid 100% ● Error 0% ● Empty 0% 10 distinct, 7 unique	● Valid 100% ● Error 0% ● Empty 0% 8 distinct, 4 unique	● Valid 100% ● Error 0% ● Empty 0% 3 distinct, 1 unique	● Valid 100% ● Error 0% ● Empty 0% 13 distinct, 12 unique	● Valid 100% ● Error 0% ● Empty 0% 1 distinct, 0 unique	● Valid ● Error ● Empty 14 distinc
1	1	14	12	9	433243547.jpeg	.jpeg	https://pri
2	2	19	12	9	Site-0.jpeg	.jpeg	https://pri
3	3	22	18	10	Site-0.jpeg	.jpeg	https://pri
4	4	32	38	6	images-0.jpeg	.jpeg	https://pri
5	5	32	38	6	images (2)-1.jpeg	.jpeg	https://pri
6	6	33	38	6	images (3)-0.jpeg	.jpeg	https://pri
7	7	36	36	6	images (1)-0.jpeg	.jpeg	https://pri
8	8	38	50	6	Construction-site-plans-1024x576-0.jpeg	.jpeg	https://pri
9	9	42	29	6	Core 3-0.jpeg	.jpeg	https://pri
10	10	42	29	6	Core 2-1.jpeg	.jpeg	https://pri
11	11	42	29	6	Core 1-0.jpeg	.jpeg	https://pri
12	12	58	59	6	images (5)-0.jpeg	.jpeg	https://pri
13	13	58	59	6	images (4)-1.jpeg	.jpeg	https://pri
14	14	62	68	9	dataicon-0.jpeg	.jpeg	https://pri

11 COLUMNS, 14 ROWS   Column profiling based on top 1000 rows

The reason to check Column number is to know the position of the **Key** and the **Binary** for the ***FunctionToSplitBinary*** function to split the binary into 30K rows against the Key.

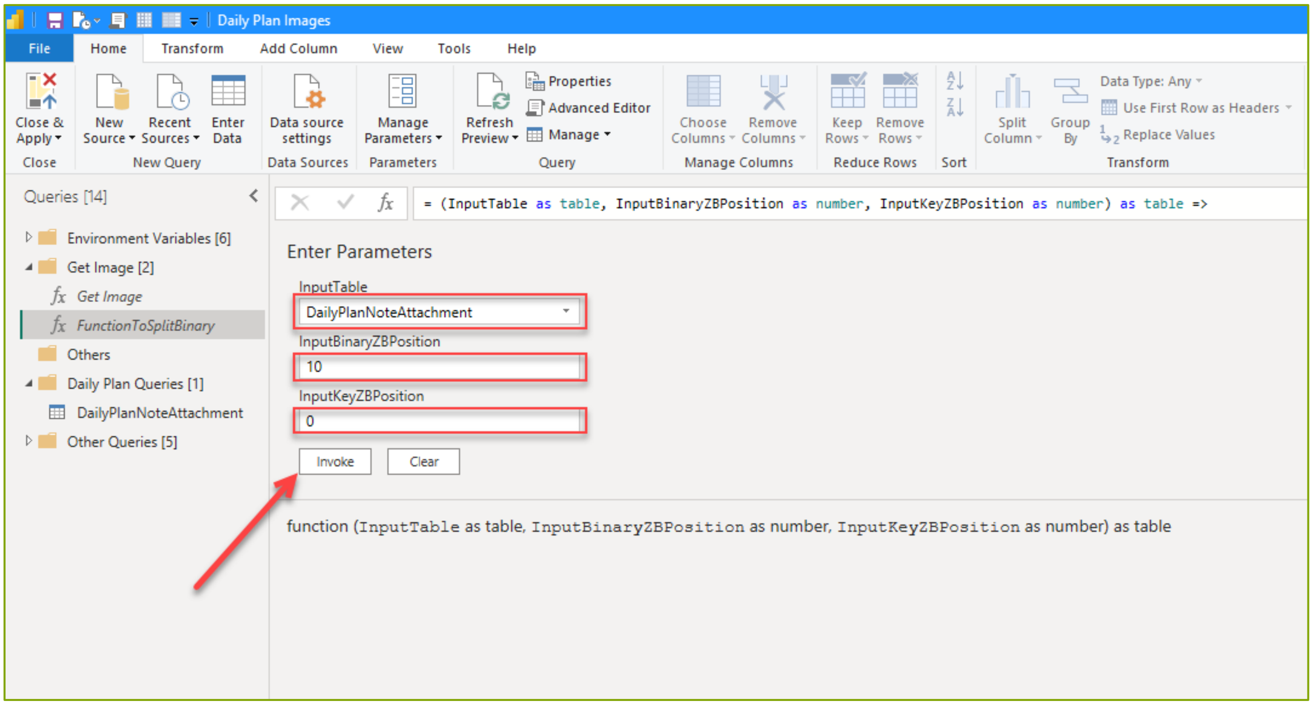
Also, the reason why this split operation must be done is because of the Power Query's 32K limitation on the binary code.

IMPORTANT: The position of the columns in Power Query is Zero base.

In this example of getting the images captured against "Notes" in Daily Plans, the position of the Key and Binary is the following.

	Position in the table
Key	0
Binary	10

Step 23: Click *FunctionToSplitBinary and enter the values as shown below and click Invoke*



```
= (InputTable as table, InputBinaryZBPosition as number, InputKeyZBPosition as number) as table =>
```

Enter Parameters

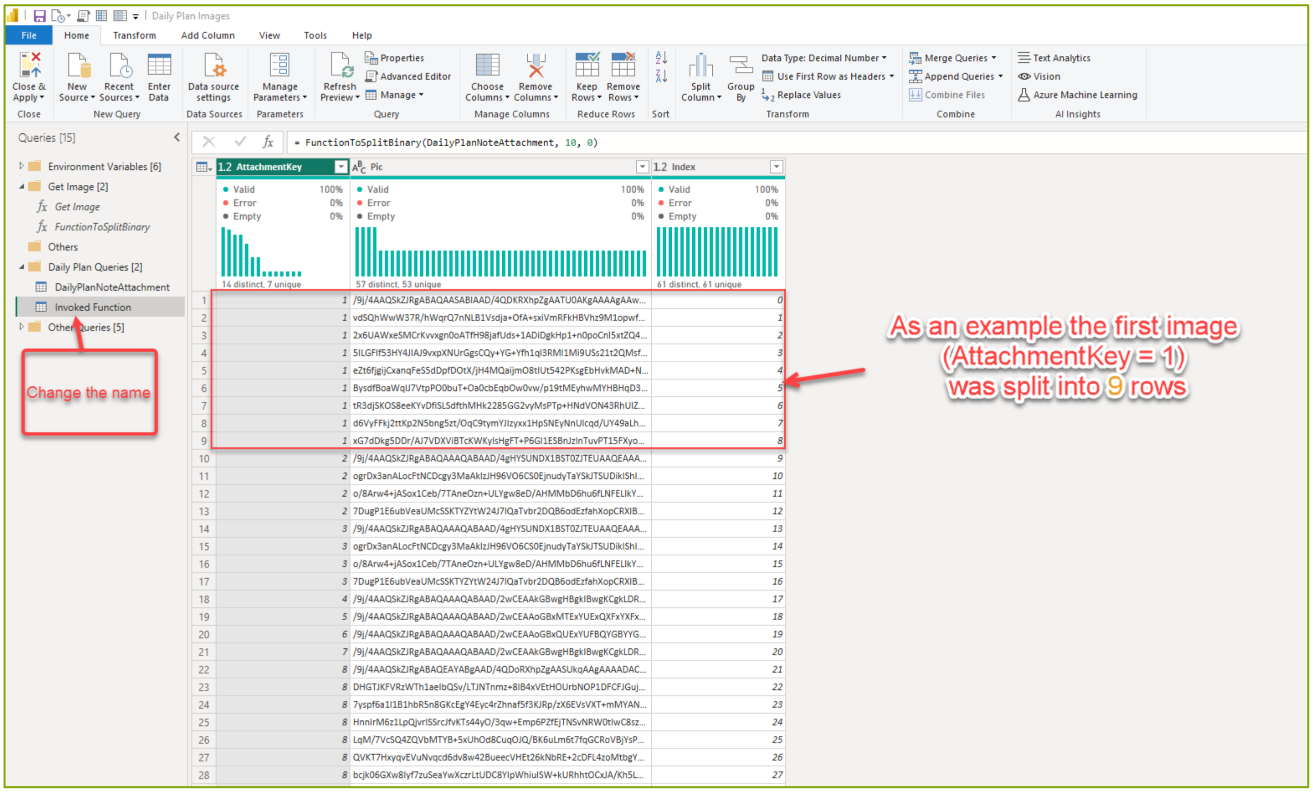
InputTable: DailyPlanNoteAttachment

InputBinaryZBPosition: 10

InputKeyZBPosition: 0

function (InputTable as table, InputBinaryZBPosition as number, InputKeyZBPosition as number) as table

Step 24: Change the name of the query NoteAttachments (OPTIONAL)

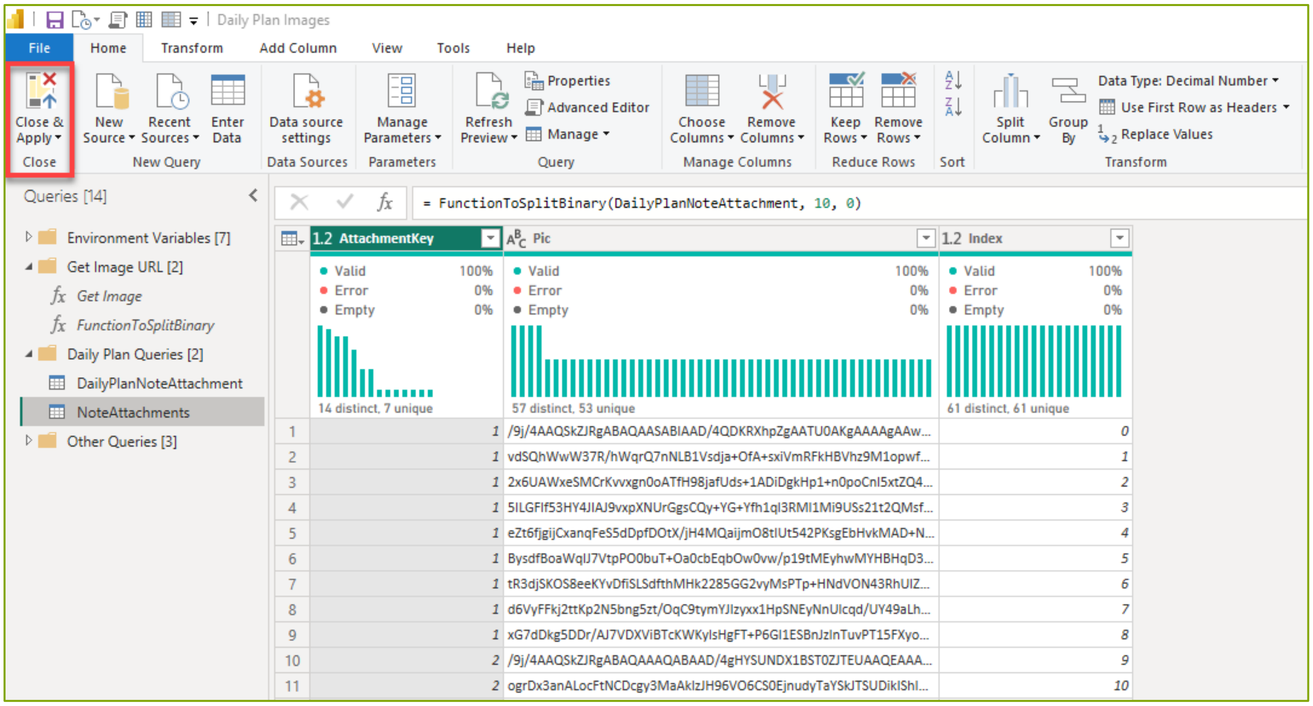


```
= FunctionToSplitBinary(DailyPlanNoteAttachment, 10, 0)
```

	1.2 AttachmentKey	A^B_C Pic	1.2 Index
	● Valid 100% ● Error 0% ● Empty 0% 14 distinct, 7 unique	● Valid 100% ● Error 0% ● Empty 0% 57 distinct, 53 unique	● Valid 100% ● Error 0% ● Empty 0% 61 distinct, 61 unique
1	1	/9j/4AAQSkZJRgABAQAASABIAAD/4QDKRXhpZgAATU0AKgAAAAgAAw...	0
2	1	vdSQhWwW37R/hWqrQ7nNLB1Vsdja+OfA+sxiVmRFkHBVhz9M1opwf...	1
3	1	2x6UAWxeSMCrKvvxgn0oATfH98jafUds+1ADiDgkHp1+n0poCnI5xtZQ4...	2
4	1	5ILGFIf53HY4JIAJ9vxpXNUrGgsCQy+YG+Yfh1qI3RMI1Mi9USs21t2QMsf...	3
5	1	eZt6fjgijCxanqFeS5dDpfDOtX/jH4MQaijmO8tIUt542PKsgEbHvkMAD+N...	4
6	1	BysdfBoaWqIJ7VtpPO0buT+Oa0cbEqbOw0vw/p19tMEyhwMYHBHqD3...	5
7	1	tR3djSKOS8eeKYvDfiSLSdfthMHk2285GG2vyMsPTp+HNdVON43RhUIZ...	6
8	1	d6VyFFkj2ttKp2N5bng5zt/OqC9tymYJlzyxx1HpSNEyNnUlcqd/UY49aLh...	7
9	1	xG7dDkg5DDr/AJ7VDXViBTcKWKylsHgFT+P6GI1ESBnJzInTuvPT15FXyo...	8
10	2	/9j/4AAQSkZJRgABAQAAAQABAAD/4gHYSUNDX1BST0ZJTEUAAQEAAA...	9
11	2	ogrDx3anALocFtNCDcgy3MaAkIzJH96VO6CS0EjnudyTaYSkJTSUDikIShI...	10
12	2	o/8Arw4+jASox1Ceb/7TAneOzn+ULYgw8eD/AHMMbD6hu6fLNFELIkY...	11
13	2	7DugP1E6ubVeaUMcSSKTYZYtW24J7IQaTvbr2DQB6odEzfahXopCRXIB...	12
14	3	/9j/4AAQSkZJRgABAQAAAQABAAD/4gHYSUNDX1BST0ZJTEUAAQEAAA...	13
15	3	ogrDx3anALocFtNCDcgy3MaAkIzJH96VO6CS0EjnudyTaYSkJTSUDikIShI...	14
16	3	o/8Arw4+jASox1Ceb/7TAneOzn+ULYgw8eD/AHMMbD6hu6fLNFELIkY...	15
17	3	7DugP1E6ubVeaUMcSSKTYZYtW24J7IQaTvbr2DQB6odEzfahXopCRXIB...	16
18	4	/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBwgHBgkIBwgKCgkLDR...	17
19	5	/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoGBxMTExYUExQXFxYXFx...	18
20	6	/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoGBxQUExYUFBQYGBYYG...	19
21	7	/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBwgHBgkIBwgKCgkLDR...	20
22	8	/9j/4AAQSkZJRgABAQEAYABgAAD/4QDoRXhpZgAASUkqAAgAAAADAC...	21
23	8	DHGTJKFVRzWTh1aeIbQSv/LTJNTnmz+8IB4xVEtHOUrbNOP1DFCFJGuj...	22
24	8	7yspf6a1I1B1hbR5n8GKcEgY4Eyc4rZhnaf5f3KJRp/zX6EVsVXT+mMYAN...	23
25	8	HnnIrM6z1LpQjvrISSrcJfvKTs44yO/3qw+Emp6PZfEjTNSvNRW0tIwC8sz...	24
26	8	LqM/7VcSQ4ZQVbMTYB+5xUhOd8CuqOJQ/BK6uLm6t7fqGCRoVBjYsP...	25
27	8	QVKT7HxyqvEVuNvqcd6dv8w42BueecVHEt26kNbRE+2cDFL4zoMtbgY...	26
28	8	bcjk06GXw8Iyf7zuSeaYwXczrLtUDC8YIpWhiuISW+kURhhtOCxJA/Kh5L...	27

As an example the first image (AttachmentKey = 1) was split into 9 rows

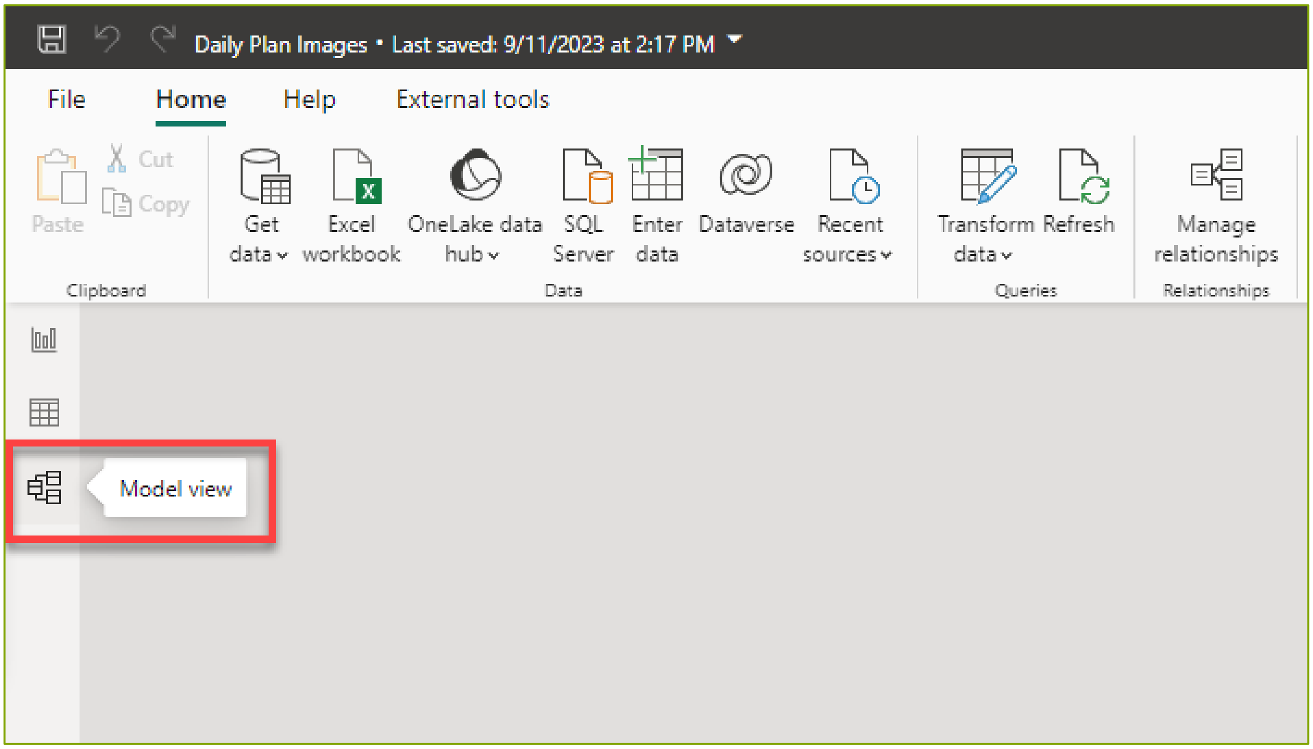
Step 25: Click Close & Load to load the tables into the data model



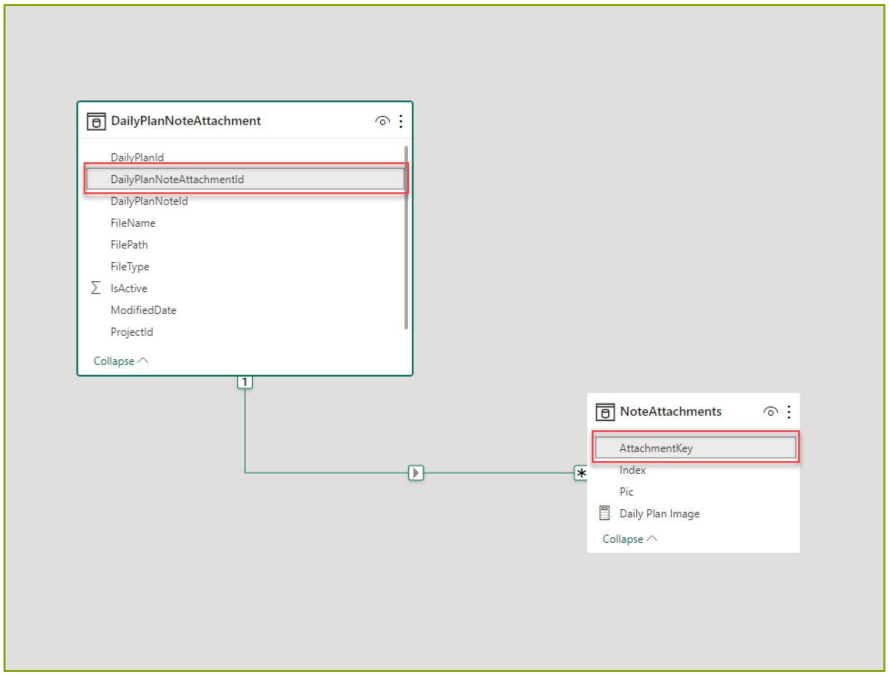
```
= FunctionToSplitBinary(DailyPlanNoteAttachment, 10, 0)
```

	1.2 AttachmentKey	ABC Pic	1.2 Index
	Valid 100% Error 0% Empty 0% 14 distinct, 7 unique	Valid 100% Error 0% Empty 0% 57 distinct, 53 unique	Valid 100% Error 0% Empty 0% 61 distinct, 61 unique
1	1	/9j/4AAQSkZJRgABAQAASABIAAD/4QDKRXhpZgAATU0AKgAAAAgAAw...	0
2	1	vdSQhWwW37R/hWqrQ7nNLB1Vsdja+OfA+sxiVmRFkHBVhz9M1opwf...	1
3	1	2x6UAWxeSMCrKvvxgn0oATfH98jafUds+1ADiDgkHp1+n0poCnl5xtZQ4...	2
4	1	5ILGFIf53HY4JIAJ9vxpXNUrGgsCQy+YG+Yfh1ql3RMI1Mi9USs21t2QMsf...	3
5	1	eZt6fjgijCxanqFeS5dDpfDOtX/jH4MQaijmO8tIUt542PKsgEbHvkMAD+N...	4
6	1	BysdfBoaWqIJ7VtpPO0buT+Oa0cbEqbOw0vw/p19tMEyhwMYHBHqD3...	5
7	1	tR3djSKOS8eeKYvDfiSLSdfthMHk2285GG2vyMsPTp+HNdVON43RhUIZ...	6
8	1	d6VyFFkj2ttKp2N5bng5zt/OqC9tymYJlzyxx1HpSNEyNnUlcqd/UY49aLh...	7
9	1	xG7dDkg5DDr/AJ7VDXViBTcKWKylsHgFT+P6GI1ESBnJzlnTuvPT15FXyo...	8
10	2	/9j/4AAQSkZJRgABAQAAAQABAAD/4gHYSUNDX1BST0ZJTEUAAQEAAA...	9
11	2	ogrDx3anALocFtNCDcgy3MaAklzJH96VO6CS0EjnudyTaYSkJTSUDikIShl...	10

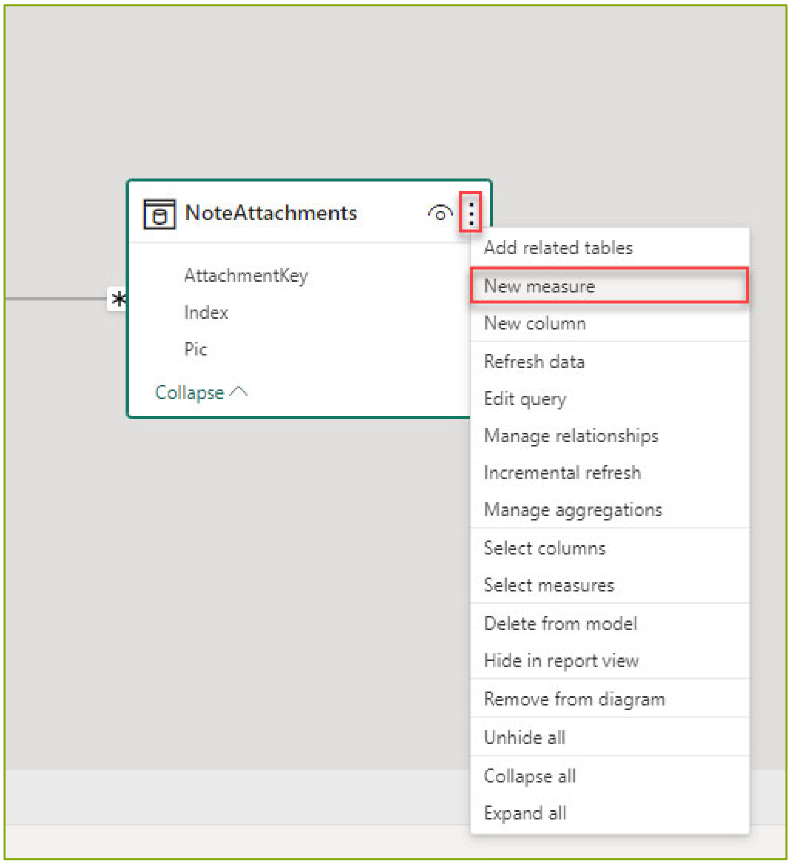
Step 26: After the data have been loaded, open the Model view



Step 27: Create a One to Many relationship between the tables as shown below.

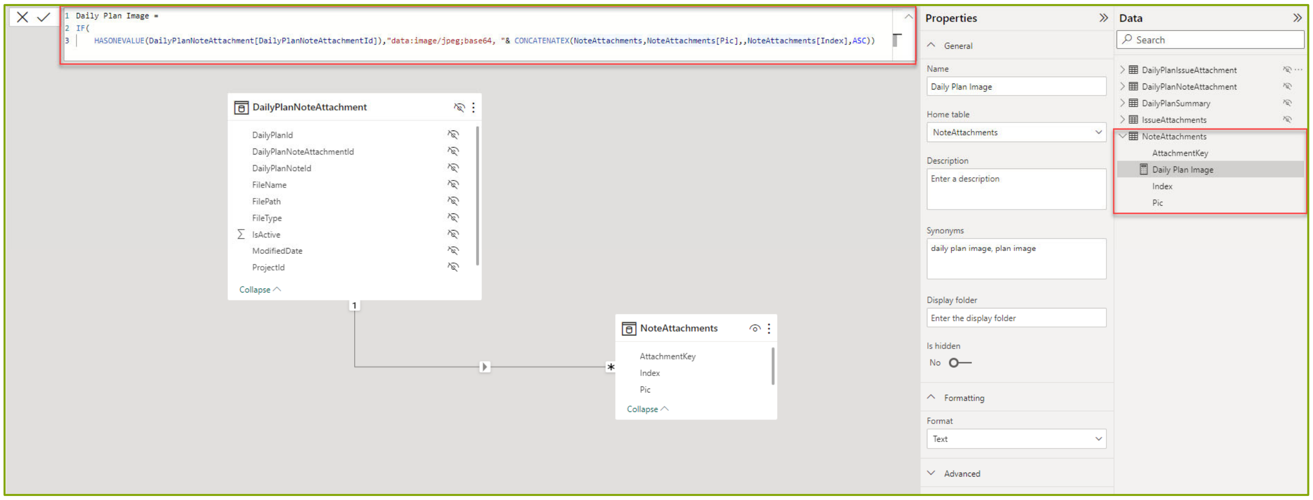


Step 28: Click on ellipses on the NoteAttachments table and select New Measure



Step 29: Enter the following code.

```
Daily Plan Image =
IF(
    HASONEVALUE(DailyPlanNoteAttachment[DailyPlanNoteAttachmentId]),"data:image/jpeg;base64, "&
CONCATENATEX(NoteAttachments,NoteAttachments[Pic],,NoteAttachments[Index],ASC))
```



```
Daily Plan Image =
IF(
    HASONEVALUE(DailyPlanNoteAttachment[DailyPlanNoteAttachmentId]),"data:image/jpeg;base64, "& CONCATENATEX(NoteAttachments,NoteAttachments[Pic],,NoteAttachments[Index],ASC))
```

Properties

General

Name: Daily Plan Image

Home table: NoteAttachments

Description: Enter a description

Synonyms: daily plan image, plan image

Display folder: Enter the display folder

Is hidden: No

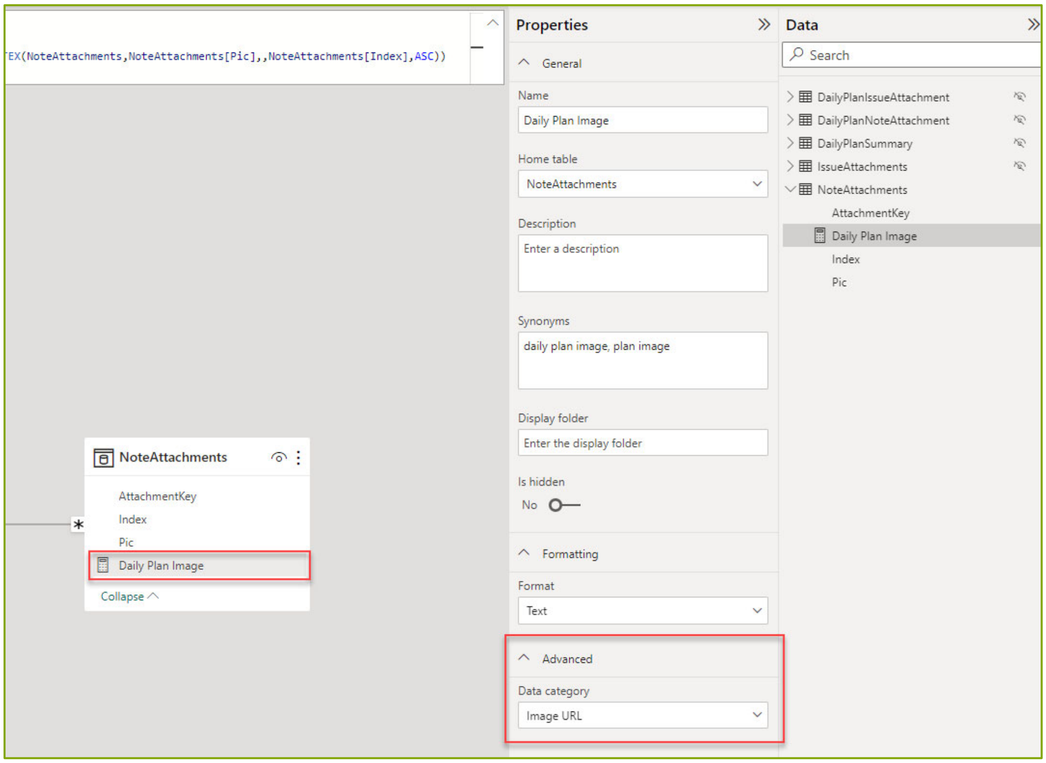
Formatting

Format: Text

Advanced

Step 30: Ensure Daily Plan Images measure is selected and then under the Properties tab expand Advanced

Step 31: Select Image URL under the Data category dropdown as shown below



```
EX(NoteAttachments,NoteAttachments[Pic],,NoteAttachments[Index],ASC))
```

Properties

General

Name: Daily Plan Image

Home table: NoteAttachments

Description: Enter a description

Synonyms: daily plan image, plan image

Display folder: Enter the display folder

Is hidden: No

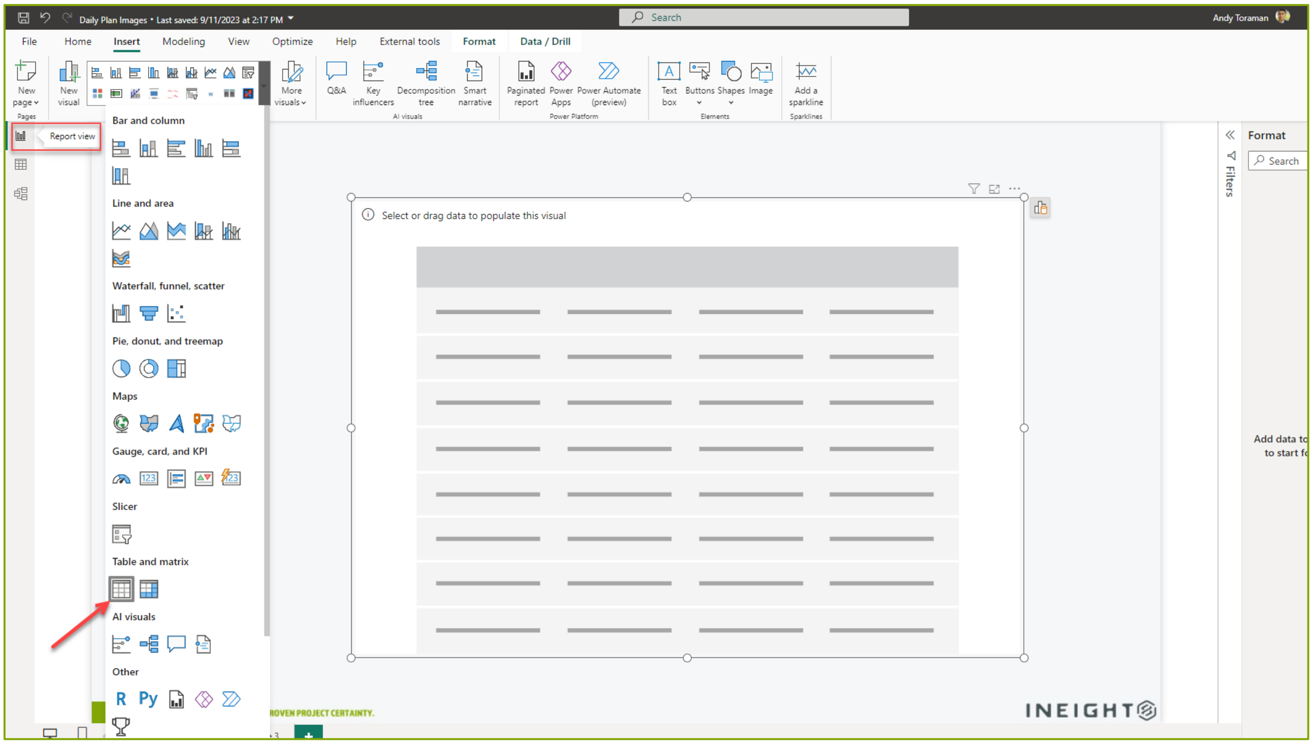
Formatting

Format: Text

Advanced

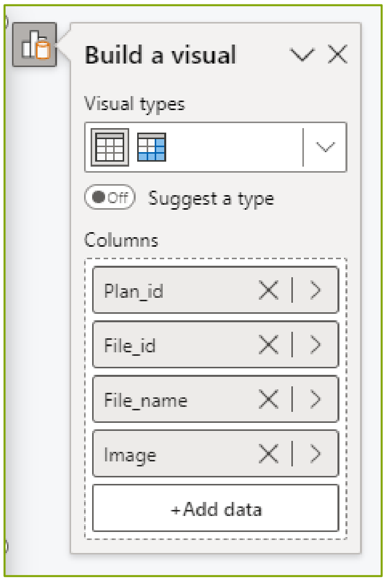
Data category: Image URL

Step 32: Open Report View and Insert a Table visual

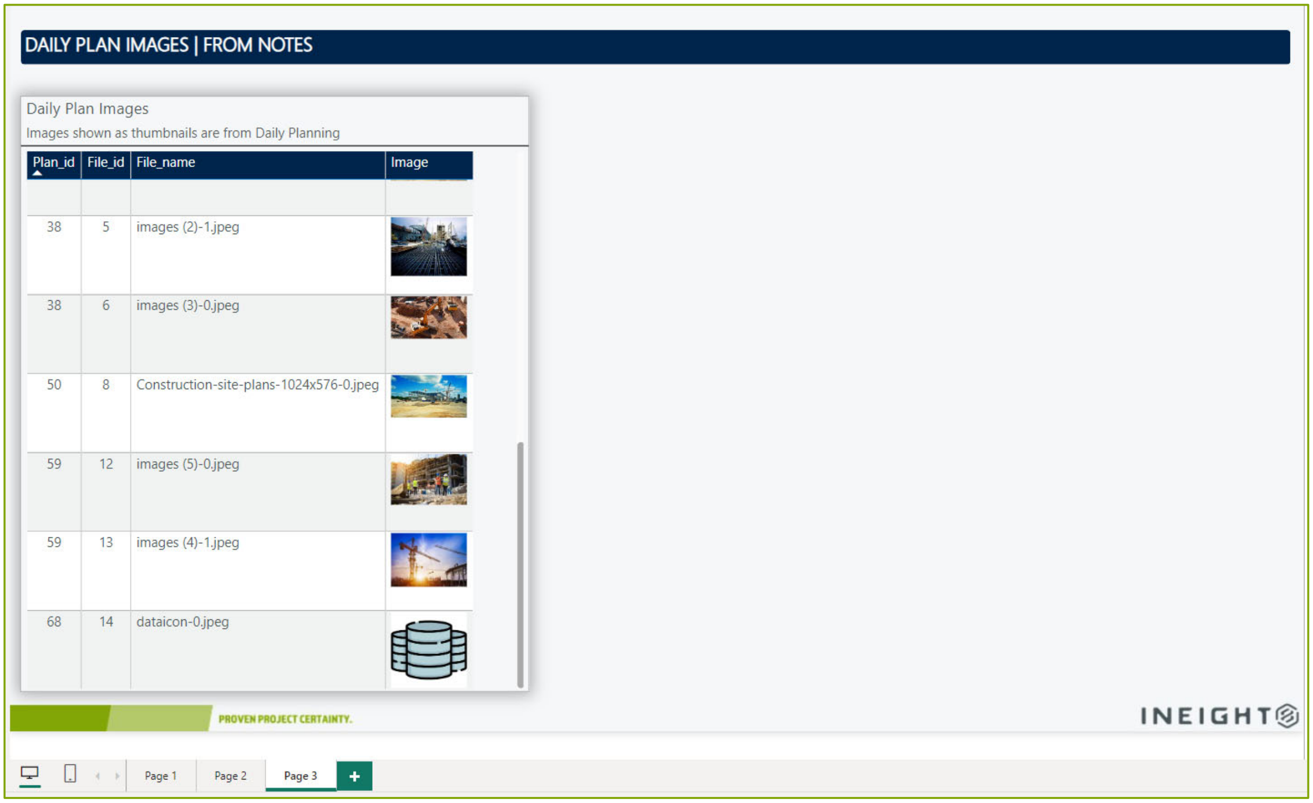


Step 33: Start adding the fields as shown below in the following order.

Order	Table	Field	Re-name (Optional)
1	DailyPlanNoteAttachment	DailyPlanId	Plan_id
2	DailyPlanNoteAttachment	DailyPlanNoteAttachmentId	File_id
3	DailyPlanNoteAttachment	FileName	File_name
4	NoteAttachments	Daily Plan Image (measure)	Image



Step 34: Images from Daily Plan Notes are now displayed



DAILY PLAN IMAGES | FROM NOTES

Daily Plan Images

Images shown as thumbnails are from Daily Planning

Plan_id	File_id	File_name	Image
38	5	images (2)-1.jpeg	
38	6	images (3)-0.jpeg	
50	8	Construction-site-plans-1024x576-0.jpeg	
59	12	images (5)-0.jpeg	
59	13	images (4)-1.jpeg	
68	14	dataicon-0.jpeg	