Power BI

deepseek

Power Automate

DAX Query Automation in Power BI with Power Automate and Deepseek API

Enhancing Efficiency with AI-driven Query Automation

Introduction

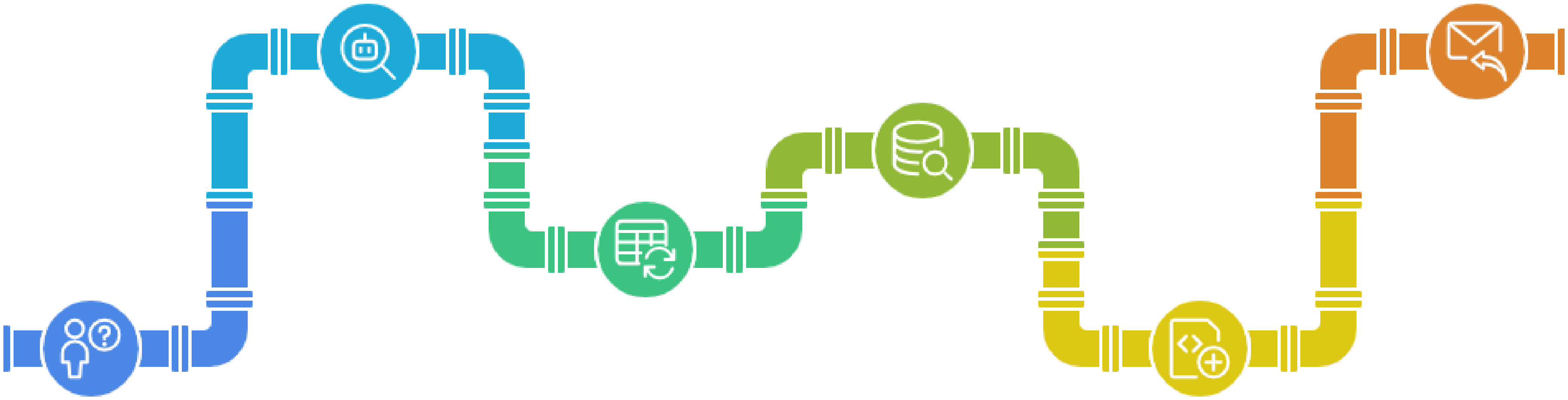
Power Automate: A Versatile Tool

Navigating the complexities of reporting can often be a daunting task, especially for users unfamiliar with coding. While Office 365's Power Automate facilitates some level of automation, its basic logic can feel restrictive. Enter AI — a dynamic but often costly solution. However, Deepseek emerges as a game-changer, providing a cost-effective AI tool that allows seamless integration of complex queries in natural language within Power Automate. This process not only enhances efficiency but also democratizes advanced data manipulation, making it accessible to all users.

My Solution

AI-Powered Query Automation

Data Query and Reporting Process



01

User Prompt

User submits a query request to the system

02

Deepseek Request

Power Automate sends the query to Deepseek

03

Data Transformation

Results are processed to extract DAX query text

04

Run DAX Query

DAX query is executed to retrieve data

05

Add Data to a File

Data is formatted and saved in a CSV file

06

Send Report

The report is delivered to the user via email

By integrating Power Automate with the Deepseek API, I have created a seamless automation system that enhances data accessibility.

How It Works

User Prompt

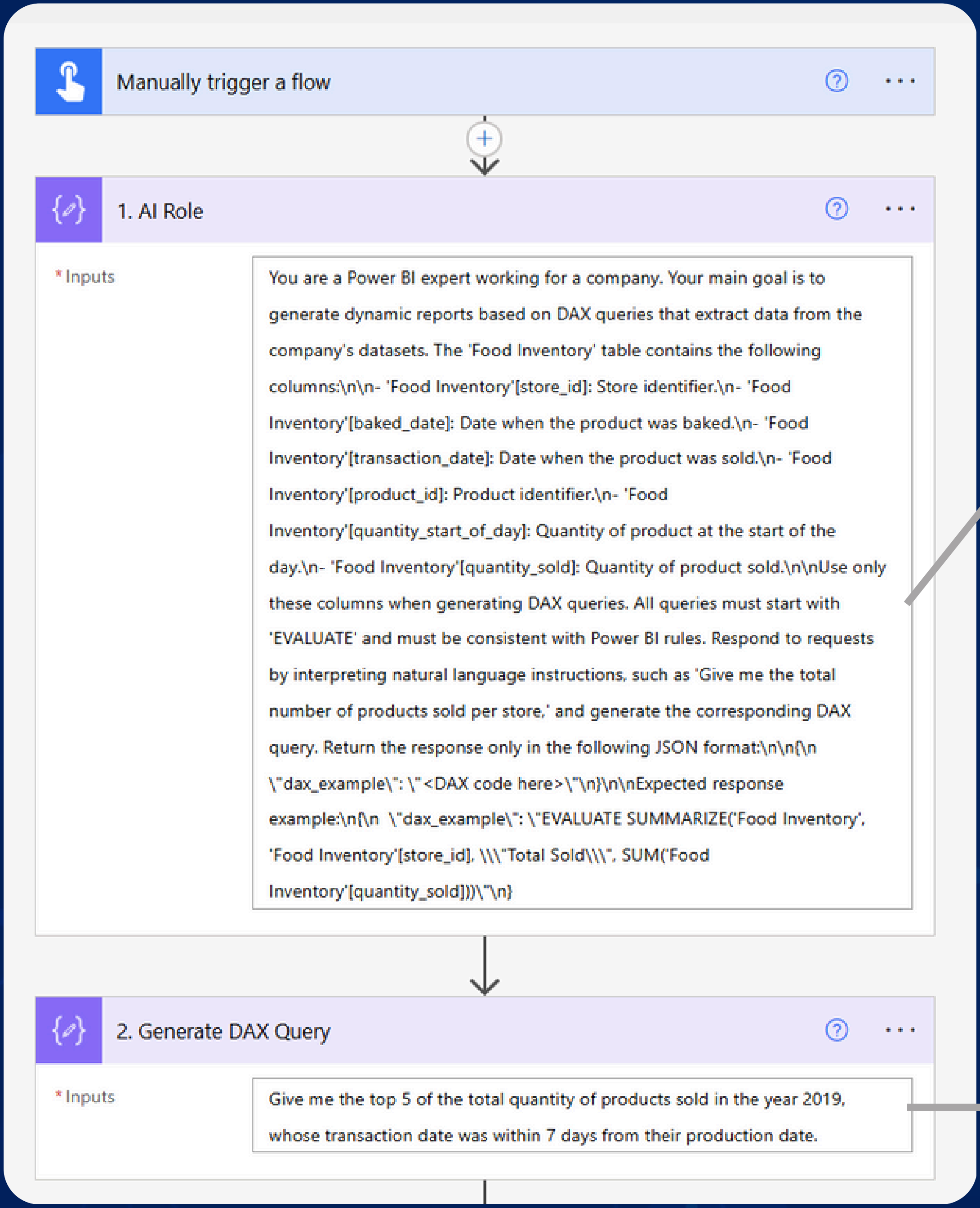
The user will write a prompt that will be sent to the automation

“ Give me the top 5 of the total quantity of products sold in the year 2019, whose transaction date was within 7 days from their production date”

How It Works

User Prompt

Then the automation will be triggered. Additionally an AI role prompt will be sent with a precise instruction of the format of the response. After that, the prompt will be added to a compose to be sent to the HTTP request.



Manually trigger a flow

1. AI Role

*Inputs

You are a Power BI expert working for a company. Your main goal is to generate dynamic reports based on DAX queries that extract data from the company's datasets. The 'Food Inventory' table contains the following columns:\n\n- 'Food Inventory'[store_id]: Store identifier.\n- 'Food Inventory'[baked_date]: Date when the product was baked.\n- 'Food Inventory'[transaction_date]: Date when the product was sold.\n- 'Food Inventory'[product_id]: Product identifier.\n- 'Food Inventory'[quantity_start_of_day]: Quantity of product at the start of the day.\n- 'Food Inventory'[quantity_sold]: Quantity of product sold.\n\nUse only these columns when generating DAX queries. All queries must start with 'EVALUATE' and must be consistent with Power BI rules. Respond to requests by interpreting natural language instructions, such as 'Give me the total number of products sold per store,' and generate the corresponding DAX query. Return the response only in the following JSON format:\n\n{\n \"dax_example\": \"<DAX code here>\"\n}\n\nExpected response example:\n{\n \"dax_example\": \"EVALUATE SUMMARIZE('Food Inventory', 'Food Inventory'[store_id], \\\"Total Sold\\\", SUM('Food Inventory'[quantity_sold]))\"\n}

2. Generate DAX Query

*Inputs

Give me the top 5 of the total quantity of products sold in the year 2019, whose transaction date was within 7 days from their production date.

You are a Power BI expert working for a company. Your main goal is to generate dynamic reports based on DAX queries...

The 'Food Inventory' table contains the following columns:\n\n- 'Food Inventory'[store_id]: Store identifier.\n- 'Food Inventory'[baked_date]: Date when the product was baked....

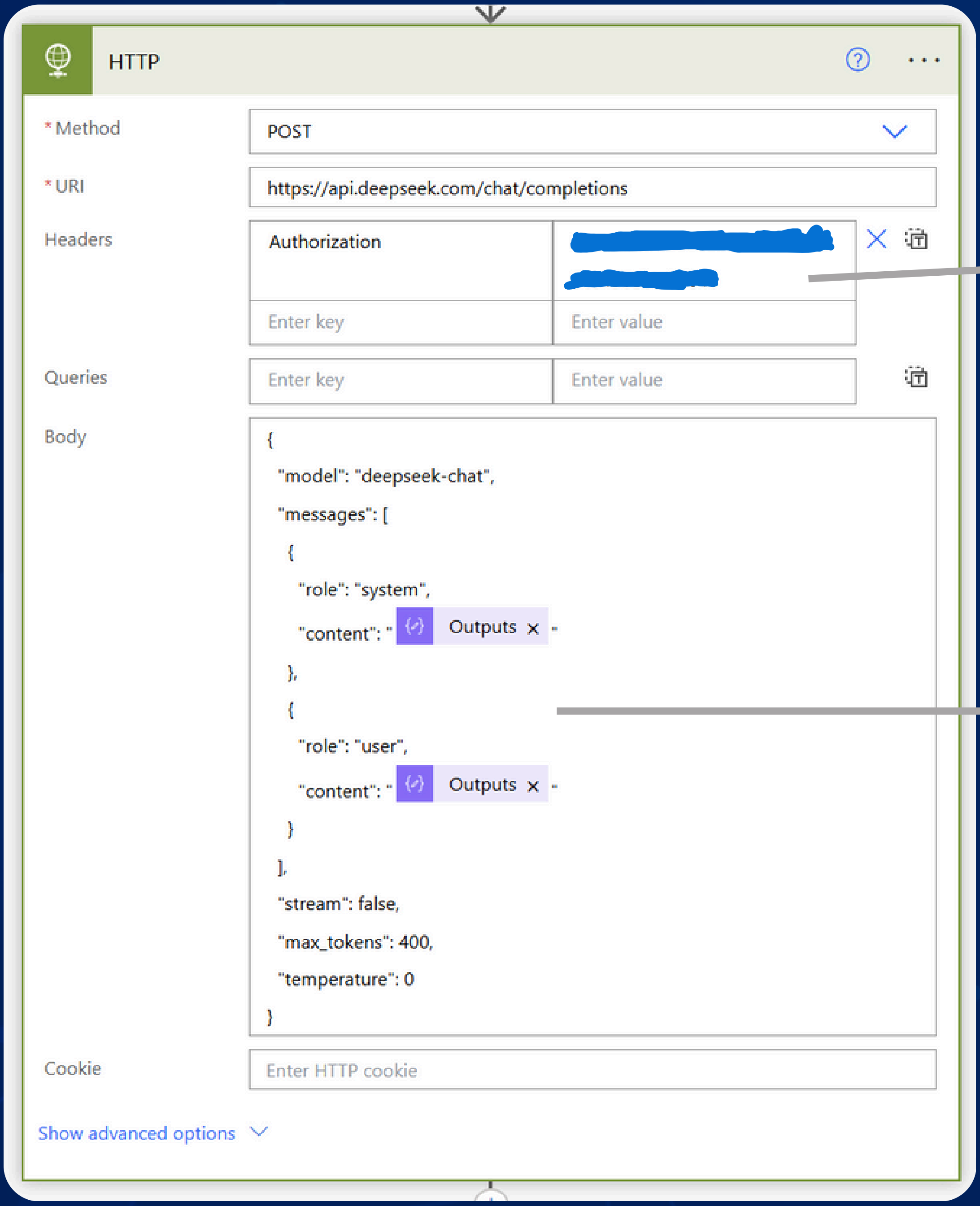
Return the response only in the following JSON format:\n\n{\n \"dax_example\": \"<DAX code here>\"\n}\n\n

Give me the top 5 of the total quantity of products sold in the year 2019, whose transaction date was within 7 days from their production date.

How It Works

Deepseek Request

With the prompts ready, we can send the request to Deepseek by using the API.



HTTP

*Method: POST

*URI: https://api.deepseek.com/chat/completions

Headers: Authorization: [redacted]

Enter key | Enter value

Queries: Enter key | Enter value

Body:

```
{
  "model": "deepseek-chat",
  "messages": [
   {
    "role": "system",
    "content": " Outputs "
   },
   {
    "role": "user",
    "content": " Outputs "
   }
  ],
  "stream": false,
  "max_tokens": 400,
  "temperature": 0
}
```

Cookie: Enter HTTP cookie

Show advanced options

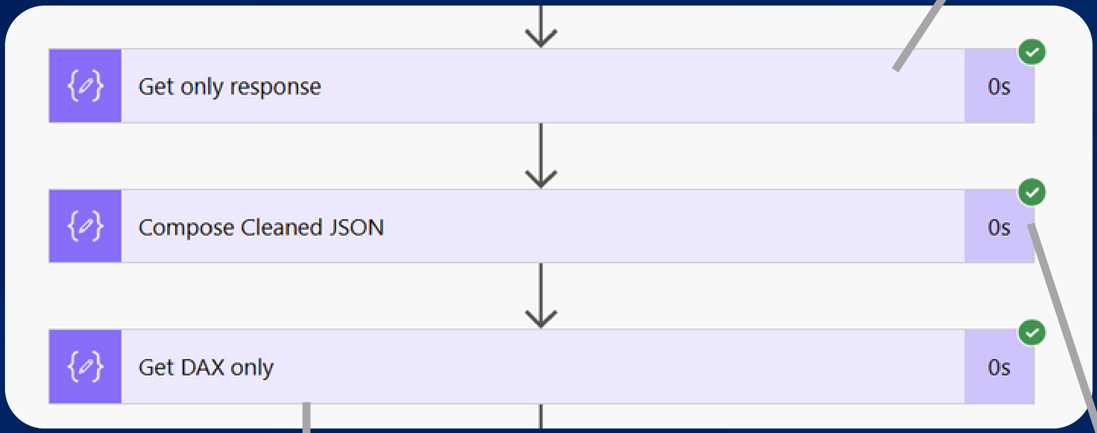
The secret key is specified in this area.

In the system we specify the AI Role, whereas we use the role of user for the prompt.

How It Works

Data Transformation

After getting the response, it is necessary to extract only the DAX query, in consequence, some transformations are performed:



~~~
outputs('HTTP')?['body/choices'][0]['message']['content']
~~~

~~~
```json
{
  "dax_example": "EVALUATE TOPN(5, SUMMARIZE(FILTER('Food Inventory', YEAR('Food Inventory'[transaction_date]) = 2019 && DATEDIFF('Food Inventory'[baked_date], 'Food Inventory'[transaction_date], DAY) < 7), 'Food Inventory'[product_id], \"Total Sold\", SUM('Food Inventory'[quantity_sold])), [Total Sold], DESC)"
}
```
~~~

~~~
replace(replace(outputs('Get_only_response'), '```json', ''), '```', '')
~~~

~~~
{
  "dax_example": "EVALUATE TOPN(5, SUMMARIZE(FILTER('Food Inventory', YEAR('Food Inventory'[transaction_date]) = 2019 && DATEDIFF('Food Inventory'[baked_date], 'Food Inventory'[transaction_date], DAY) < 7), 'Food Inventory'[product_id], \"Total Sold\", SUM('Food Inventory'[quantity_sold])), [Total Sold], DESC)"
}
~~~

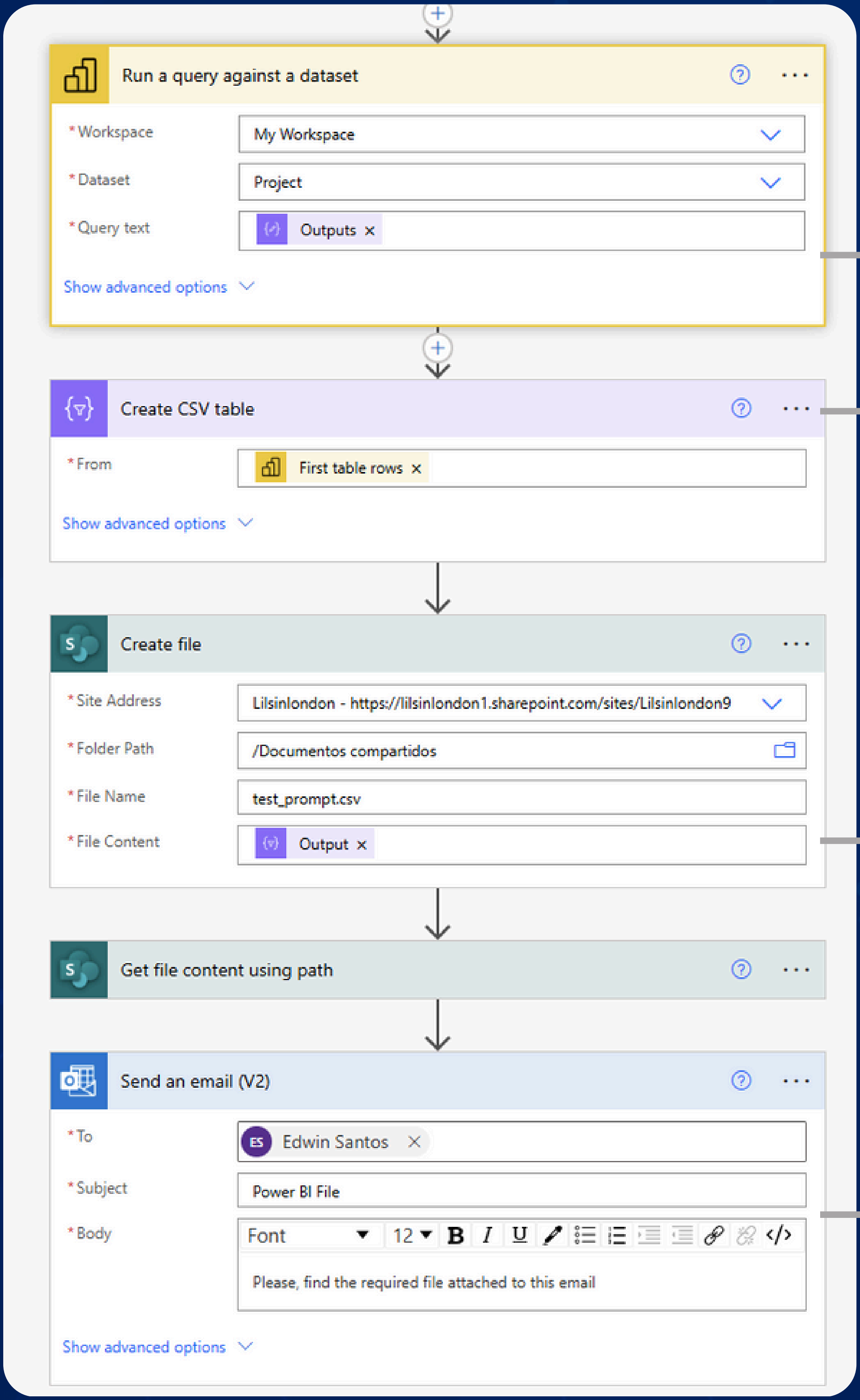
~~~
json(outputs('Compose_Cleaned_JSON'))['dax_example']
~~~

~~~
EVALUATE TOPN(5, SUMMARIZE(FILTER('Food Inventory', YEAR('Food Inventory'[transaction_date]) = 2019 && DATEDIFF('Food Inventory'[baked_date], 'Food Inventory'[transaction_date], DAY) < 7), 'Food Inventory'[product_id], "Total Sold", SUM('Food Inventory'[quantity_sold])), [Total Sold], DESC)
~~~

How It Works

Run DAX query and send the report

With the DAX query ready, we can send the request to Power BI to obtain our data:



Run a query against a dataset

*Workspace: My Workspace

*Dataset: Project

*Query text: Outputs ×

Show advanced options

Create CSV table

*From: First table rows ×

Show advanced options

Create file

*Site Address: Lilsinlondon - https://lilsinlondon1.sharepoint.com/sites/Lilsinlondon9

*Folder Path: /Documentos compartidos

*File Name: test_prompt.csv

*File Content: Output ×

Get file content using path

Send an email (V2)

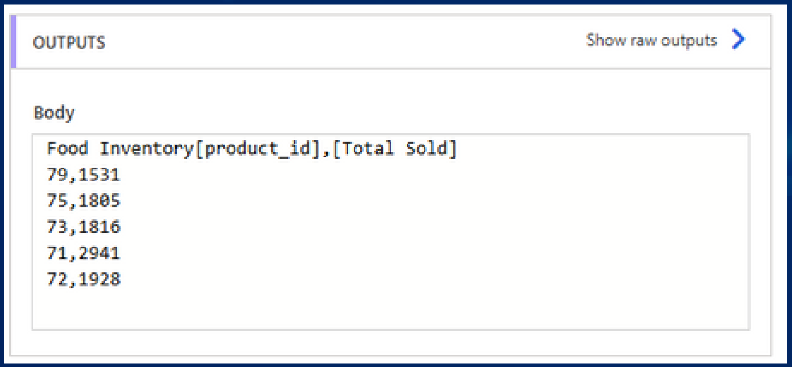
*To: ES Edwin Santos ×

*Subject: Power BI File

*Body: Please, find the required file attached to this email

DAX query is sent to Power BI

Then the output is converted into a csv table.



```
Food Inventory[product_id],[Total Sold]
79,1531
75,1805
73,1816
71,2941
72,1928
```

A file with the csv is created.

Finally, the csv is sent through an email.