Optional Lab – Explore Copilot capabilities

Lab Guide

Oct 2024

Contents

Lab Description 2
Pre-Requisite..2
Lab Tasks 2
Exercises 1: Explore Copilot for data flows 2
Task 1: Ingest a Dataset into Fabric with Data Pipelines Copilot. 2
Task 2: Ingest the Dataset via Data Pipelines to Lakehouse File Section 6
Exercise 2: Explore Copilot capabilities in Fabric notebooks 13

Lab Description

Microsoft has integrated Copilot and other generative AI features into Fabric to introduce new ways for you to transform and analyze data, generate insights, and create visualizations and reports. You must enable Copilot before you can use it. Copilot in Fabric is not available in all regions. Microsoft includes the following Copilot integrations:

- Data Factory: This integration includes tools that help you streamline workflows. The toolset can both generate and explain code for transforming data.
- Power BI: This integration includes generative AI to build reports automatically based on topics that you select or prompts that you create.

In this exercise, you will explore Copilot Data Factory capabilities.

You will explore the Power BI Copilot in a later exercise in this lab.

Note : There is no dependency on other labs execution to run this lab

Pre-Requisite

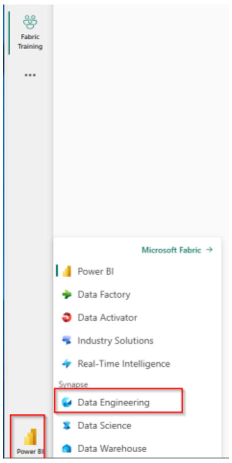
Provision a paid capacity before you proceed with this lab . Trial capacity does not support Co-Pilot features.

Lab Tasks

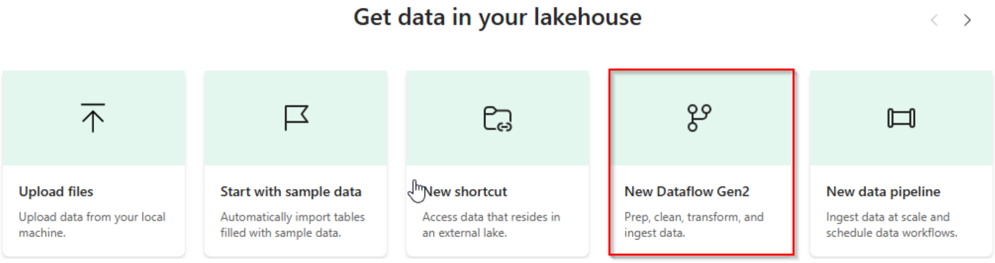
Exercises 1: Explore Copilot for data flows

Task 1: Ingest a Dataset into Fabric with Data Pipelines Copilot.

1. On the left menu, select **Power BI**, then under **Synapse**, select **Data Engineering**.



2. Select the bronze Lakehouse you have created on the previous lab.
3. Under **Get data** in your lakehouse. select **New Dataflow Gen2**.


Get data in your lakehouse

Upload files
Upload data from your local machine.

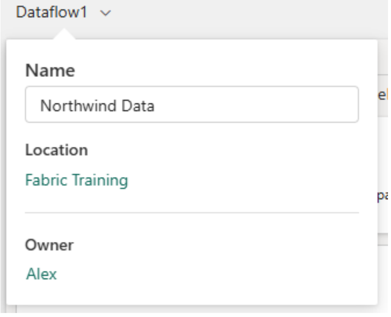
Start with sample data
Automatically import tables filled with sample data.

New shortcut
Access data that resides in an external lake.

New Dataflow Gen2
Prep, clean, transform, and ingest data.

New data pipeline
Ingest data at scale and schedule data workflows.

4. Select the **Dataflow 1** in the menu, then in the **Name** box, enter Northwind Data

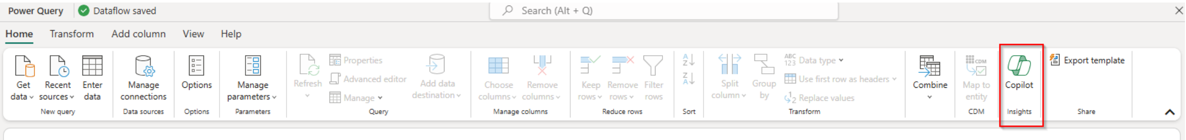


Name
Northwind Data

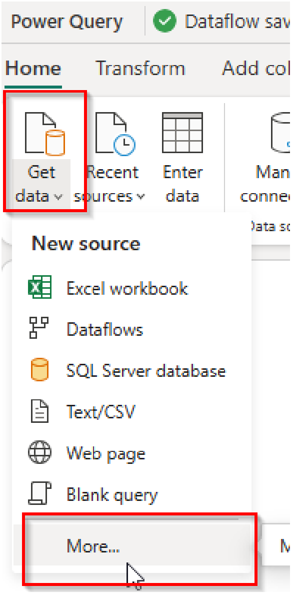
Location
Fabric Training

Owner
Alex

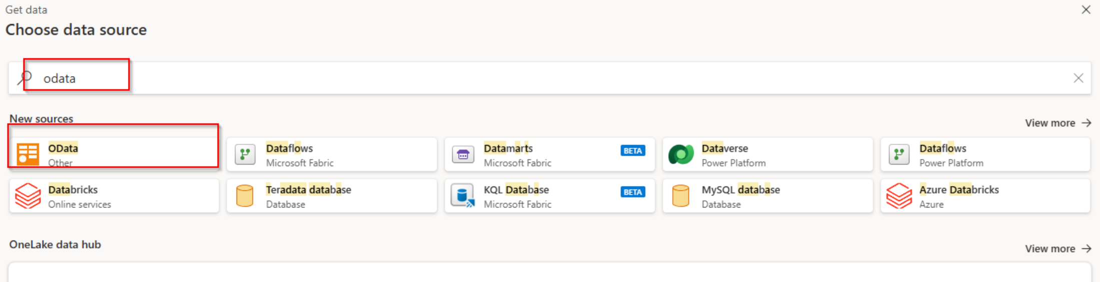
5. On the **Power Query** toolbar, on the **Home** tab, select **CoPilot**:



6. On the **Home** tab, select **Get data**, then under **New source**, select **More**:



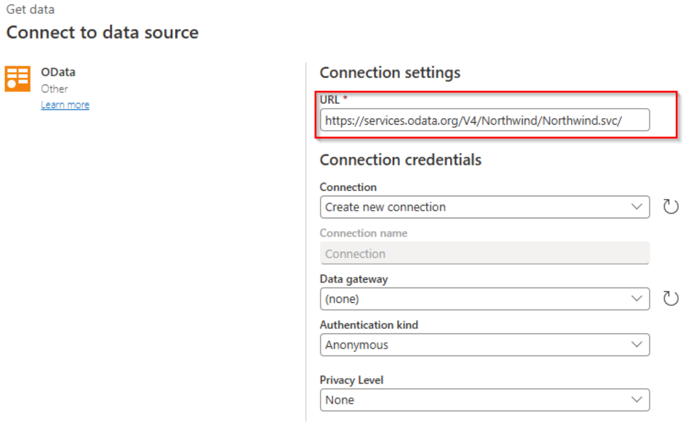
7. In the **Choose data source** box, enter odata to filter the possible data sources, then select **OData**.


Get data

Choose data source

odata

8. In the **Connect to data source** window, under **Connection settings**, in the **URL** box, enter https://services.odata.org/V4/Northwind/Northwind.svc/, then select **Next**.


Get data

Connect to data source

OData
Other
Learn more

Connection settings

URL *
https://services.odata.org/V4/Northwind/Northwind.svc/

Connection credentials

Connection: Create new connection

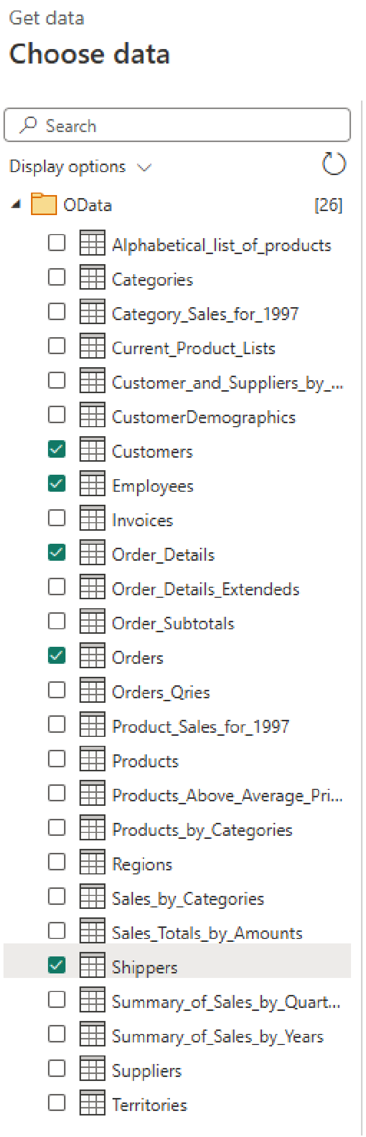
Connection name: Connection

Data gateway: (none)

Authentication kind: Anonymous

Privacy Level: None

9. On the **Choose Data** window, select **Orders** checkbox, select **Select related tables**, then select **Create**.



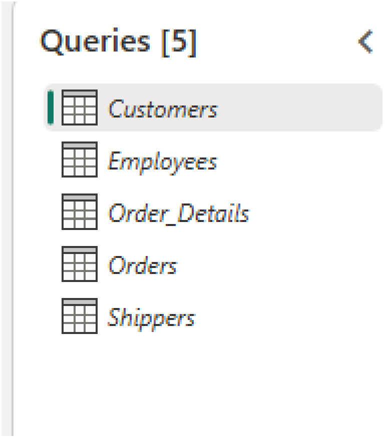
Get data

Choose data

OData [26]

- ☐ Alphabetical_list_of_products
- ☐ Categories
- ☐ Category_Sales_for_1997
- ☐ Current_Product_Lists
- ☐ Customer_and_Suppliers_by_...
- ☐ CustomerDemographics
- ☑ Customers
- ☑ Employees
- ☐ Invoices
- ☑ Order_Details
- ☐ Order_Details_Extendeds
- ☐ Order_Subtotals
- ☑ Orders
- ☐ Orders_Qries
- ☐ Product_Sales_for_1997
- ☐ Products
- ☐ Products_Above_Average_Pri...
- ☐ Products_by_Categories
- ☐ Regions
- ☐ Sales_by_Categories
- ☐ Sales_Totals_by_Amounts
- ☑ Shippers
- ☐ Summary_of_Sales_by_Quart...
- ☐ Summary_of_Sales_by_Years
- ☐ Suppliers
- ☐ Territories

The query should look like the following query.

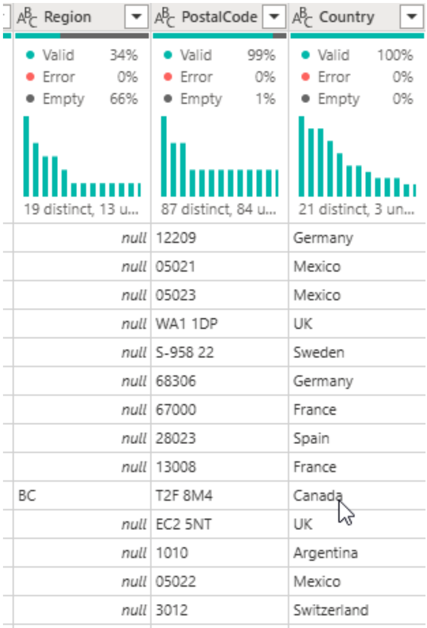


Queries [5]

- Customers
- Employees
- Order_Details
- Orders
- Shippers

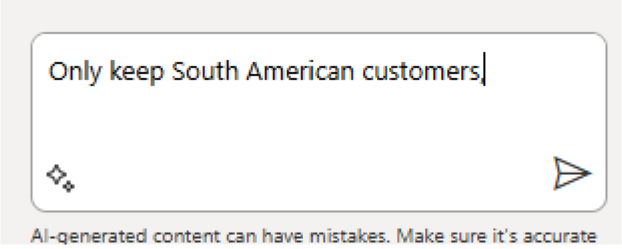
Task 2: Ingest the Dataset via Data Pipelines to Lakehouse File Section

1. In the customers table, scroll to the right and examine the Country column. Notice the countries include **Argentina** and **Mexico**.



Region	PostalCode	Country
Valid 34% Error 0% Empty 66%	Valid 99% Error 0% Empty 1%	Valid 100% Error 0% Empty 0%
19 distinct, 13 u...	87 distinct, 84 u...	21 distinct, 3 un...
null	12209	Germany
null	05021	Mexico
null	05023	Mexico
null	WA1 1DP	UK
null	S-958 22	Sweden
null	68306	Germany
null	67000	France
null	28023	Spain
null	13008	France
BC	T2F 8M4	Canada
null	EC2 5NT	UK
null	1010	Argentina
null	05022	Mexico
null	3012	Switzerland

2. In the **Copilot** pane enter *Only keep South American customers*, then select **Send**.



Only keep South American customers

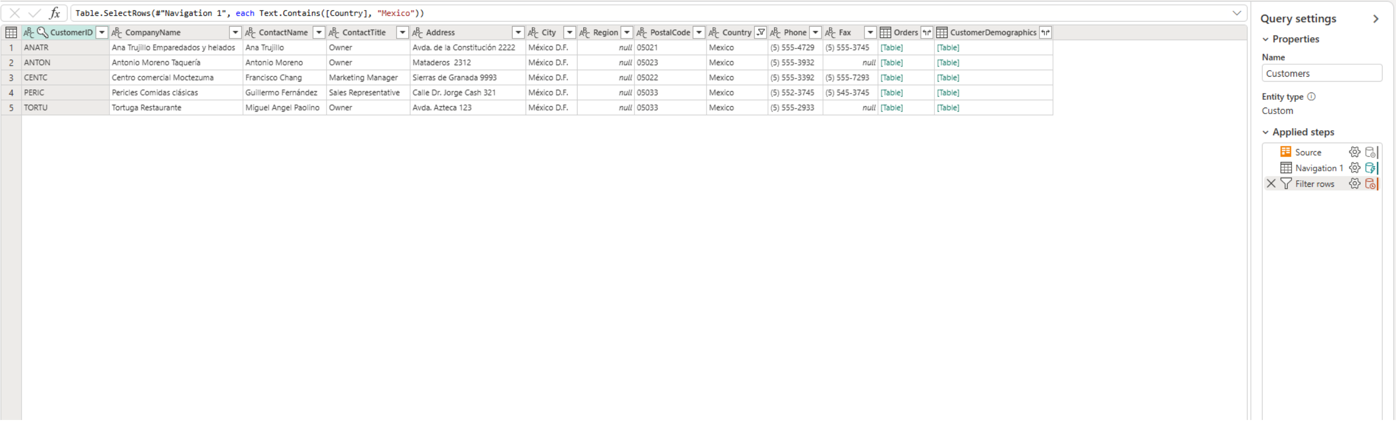
AI-generated content can have mistakes. Make sure it's accurate

Note:

Due to the nature of Copliot you may end up with differing results. You can also try *Only keep customers from South American countries*

The desired Applied Step text is :
Table.SelectRows(#"Navigation 1", each List.Contains({"Mexico", "Brazil", "Argentina", "Chile", "Peru", "Colombia", "Venezuela", "Ecuador", "Bolivia", "Paraguay", "Uruguay", "Guyana", "Suriname"}, [Country]))

3. a) As you can see, it selected Mexico only:



```
Table.SelectRows(#"Navigation 1", each Text.Contains([Country], "Mexico"))
```

	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax	Orders	CustomerDemographics
1	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	null	05021	Mexico	(5) 555-4729	(5) 555-3745	[Table]	[Table]
2	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	null	05023	Mexico	(5) 555-3932	null	[Table]	[Table]
3	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.	null	05022	Mexico	(5) 555-3392	(5) 555-7293	[Table]	[Table]
4	PERIC	Pericles Comidas clásicas	Guillermo Fernández	Sales Representative	Calle Dr. Jorge Cash 321	México D.F.	null	05033	Mexico	(5) 552-3745	(5) 545-3745	[Table]	[Table]
5	TORTU	Tortuga Restaurante	Miguel Angel Paolino	Owner	Avda. Azteca 123	México D.F.	null	05033	Mexico	(5) 555-2933	null	[Table]	[Table]

Query settings

Properties

Name: Customers

Entity type: Custom

Applied steps

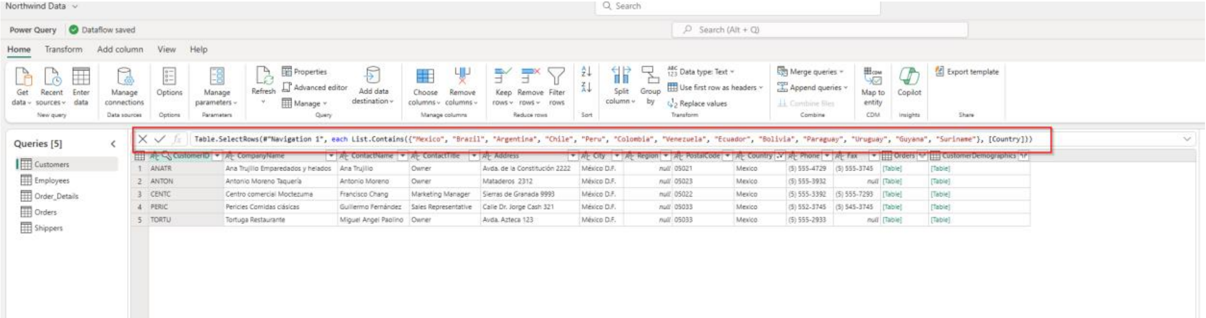
- Source
- Navigation 1
- Filter rows

You can undo the step by clicking on Undo

Afterwards type: *Only keep South American customers*

b) If this step returns Mexico only, then insert this step in the transformation tab:

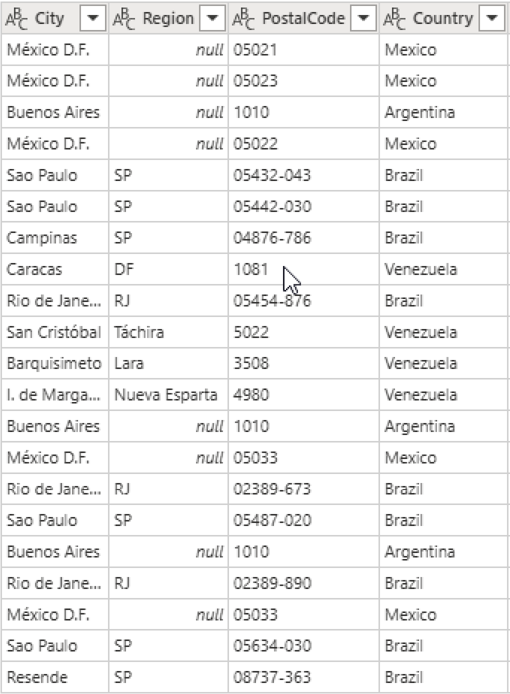
Table.SelectRows(#"Navigation 1", each List.Contains({"Mexico", "Brazil", "Argentina", "Chile", "Peru", "Colombia", "Venezuela", "Ecuador", "Bolivia", "Paraguay", "Uruguay", "Guyana", "Suriname"}, [Country]))



```
Table.SelectRows(#"Navigation 1", each List.Contains({"Mexico", "Brazil", "Argentina", "Chile", "Peru", "Colombia", "Venezuela", "Ecuador", "Bolivia", "Paraguay", "Uruguay", "Guyana", "Suriname"}, [Country]))
```

	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax	Orders	CustomerDemographics
1	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	null	05021	Mexico	(5) 555-4729	(5) 555-3745	[Table]	[Table]
2	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	null	05023	Mexico	(5) 555-3932	null	[Table]	[Table]
3	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.	null	05022	Mexico	(5) 555-3392	(5) 555-7293	[Table]	[Table]
4	PERIC	Pericles Comidas clásicas	Guillermo Fernández	Sales Representative	Calle Dr. Jorge Cash 321	México D.F.	null	05033	Mexico	(5) 552-3745	(5) 545-3745	[Table]	[Table]
5	TORTU	Tortuga Restaurante	Miguel Angel Paolino	Owner	Avda. Azteca 123	México D.F.	null	05033	Mexico	(5) 555-2933	null	[Table]	[Table]

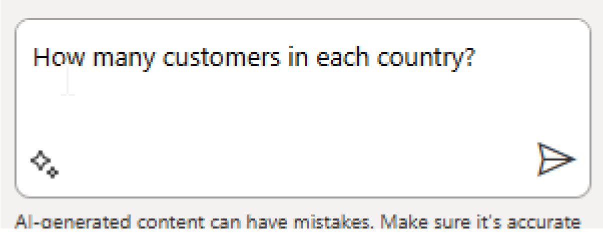
4. The Country column has been filtered and only includes customers from South America.



City	Region	PostalCode	Country
México D.F.	null	05021	Mexico
México D.F.	null	05023	Mexico
Buenos Aires	null	1010	Argentina
México D.F.	null	05022	Mexico
Sao Paulo	SP	05432-043	Brazil
Sao Paulo	SP	05442-030	Brazil
Campinas	SP	04876-786	Brazil
Caracas	DF	1081	Venezuela
Rio de Jane...	RJ	05454-876	Brazil
San Cristóbal	Táchira	5022	Venezuela
Barquisimeto	Lara	3508	Venezuela
I. de Marga...	Nueva Esparta	4980	Venezuela
Buenos Aires	null	1010	Argentina
México D.F.	null	05033	Mexico
Rio de Jane...	RJ	02389-673	Brazil
Sao Paulo	SP	05487-020	Brazil
Buenos Aires	null	1010	Argentina
Rio de Jane...	RJ	02389-890	Brazil
México D.F.	null	05033	Mexico
Sao Paulo	SP	05634-030	Brazil
Resende	SP	08737-363	Brazil

5. In the **Copilot** pane, select **Undo** to revert the changes as we want to use the Copilot against the whole dataset.

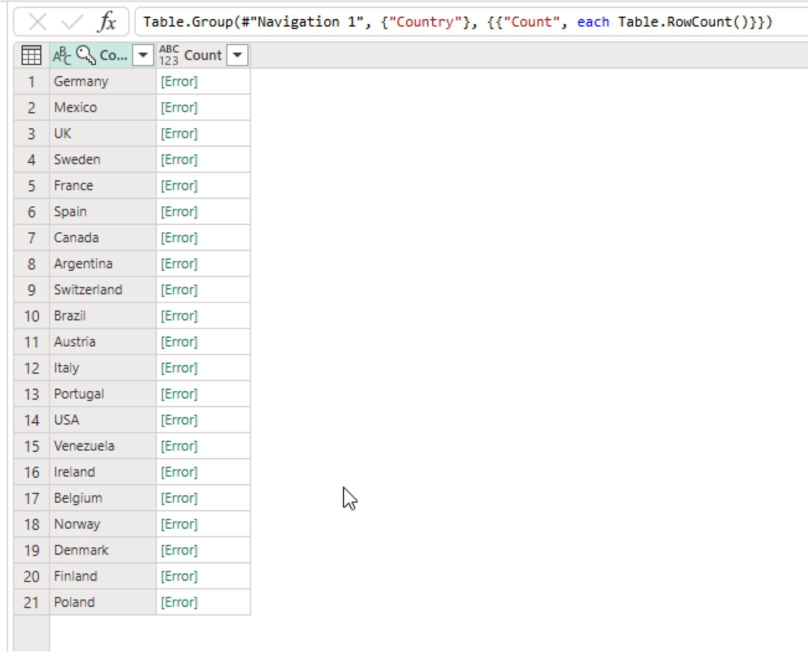
6. In the **Copilot** pane enter *How many customers in each country?*, then select **Send**.



How many customers in each country?

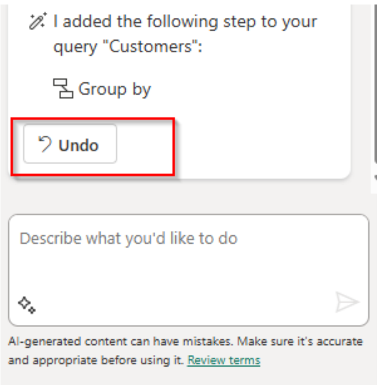
AI-generated content can have mistakes. Make sure it's accurate

! Sometimes, due to the nature of Copliot you may end up with different results or errors. Click on Undo in the Copilot chat:



```
Table.Group(#"Navigation 1", {"Country"}, {{"Count", each Table.RowCount()}})
```

	Co...	Count
1	Germany	[Error]
2	Mexico	[Error]
3	UK	[Error]
4	Sweden	[Error]
5	France	[Error]
6	Spain	[Error]
7	Canada	[Error]
8	Argentina	[Error]
9	Switzerland	[Error]
10	Brazil	[Error]
11	Austria	[Error]
12	Italy	[Error]
13	Portugal	[Error]
14	USA	[Error]
15	Venezuela	[Error]
16	Ireland	[Error]
17	Belgium	[Error]
18	Norway	[Error]
19	Denmark	[Error]
20	Finland	[Error]
21	Poland	[Error]



I added the following step to your query "Customers":

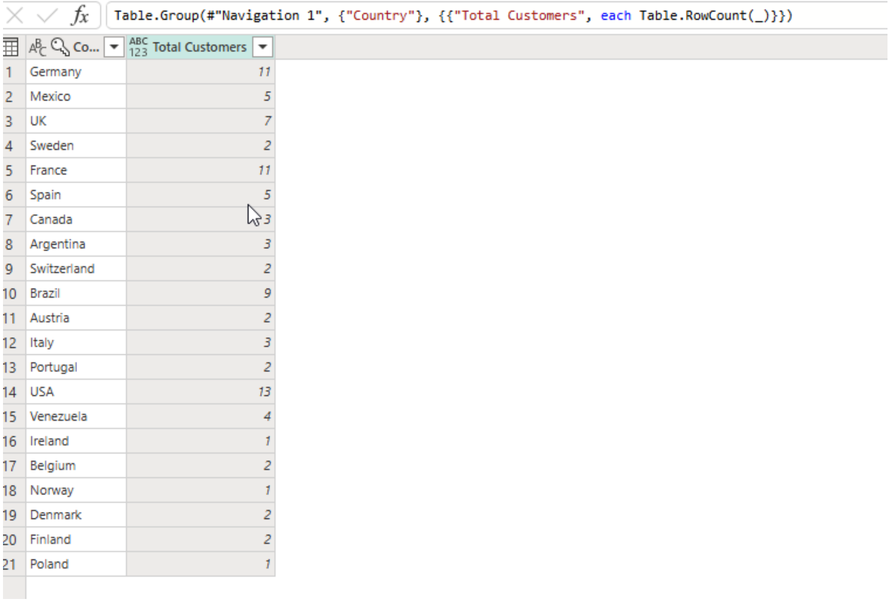
Group by

AI-generated content can have mistakes. Make sure it's accurate and appropriate before using it. Review terms

You need to have an accurate question, so you can also try:
What is the total number of customers in each country ?

The desired Applied Step text is :
Table.Group(#"Navigation 1", {"Country"}, {{"Total Customers", each Table.RowCount(_)}})

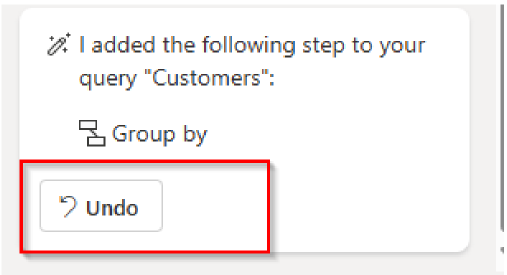
7. The query outputs a list displaying the number of customers per country.



```
Table.Group(#"Navigation 1", {"Country"}, {{"Total Customers", each Table.RowCount(_)}})
```

	Co...	Total Customers
1	Germany	11
2	Mexico	5
3	UK	7
4	Sweden	2
5	France	11
6	Spain	5
7	Canada	3
8	Argentina	3
9	Switzerland	2
10	Brazil	9
11	Austria	2
12	Italy	3
13	Portugal	2
14	USA	13
15	Venezuela	4
16	Ireland	1
17	Belgium	2
18	Norway	1
19	Denmark	2
20	Finland	2
21	Poland	1

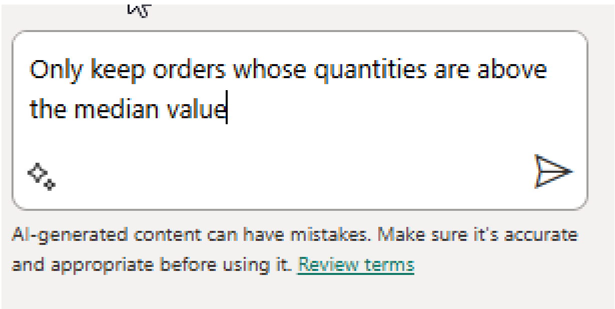
8. In the **Copilot** pane, select **Undo** to revert the changes.



I added the following step to your query "Customers":

Group by

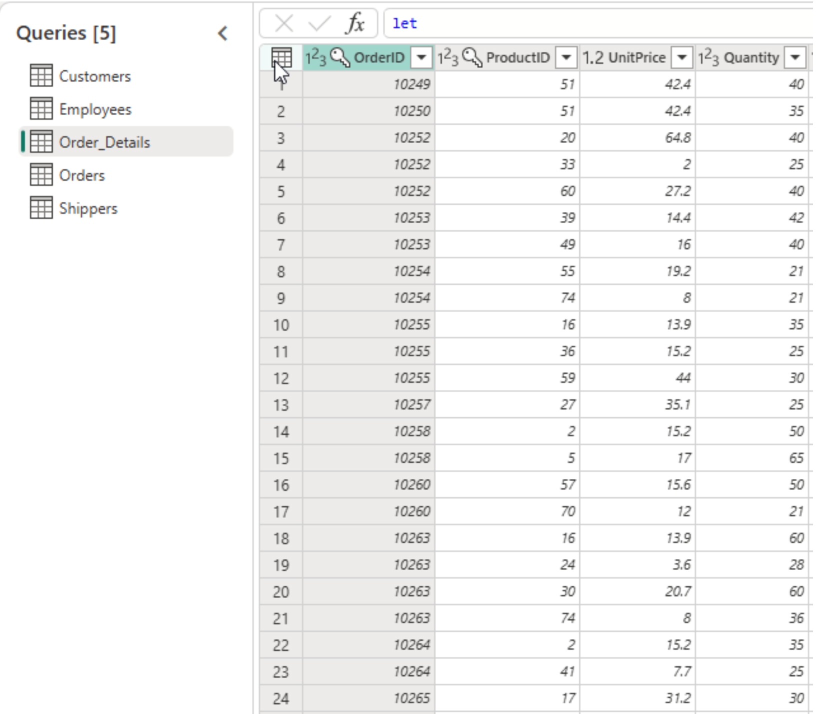
9. Select the **Order_Details** query, then in the **Copilot** pane, enter *Only keep orders whose quantities are above the median value*, then select **Send**.



Only keep orders whose quantities are above the median value

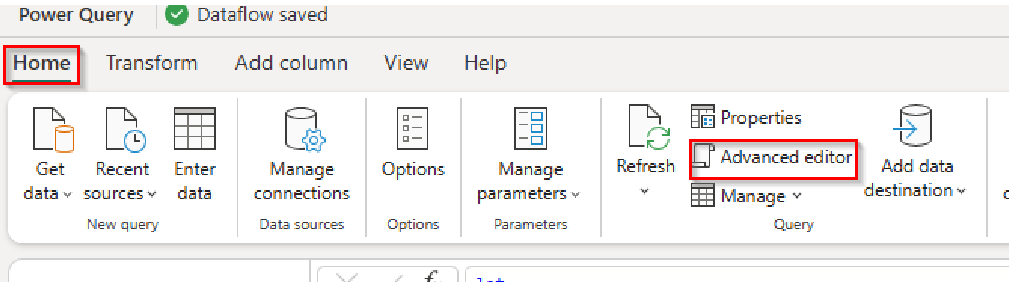
AI-generated content can have mistakes. Make sure it's accurate and appropriate before using it. Review terms

10. The Quantity Column now displays values all above 20:

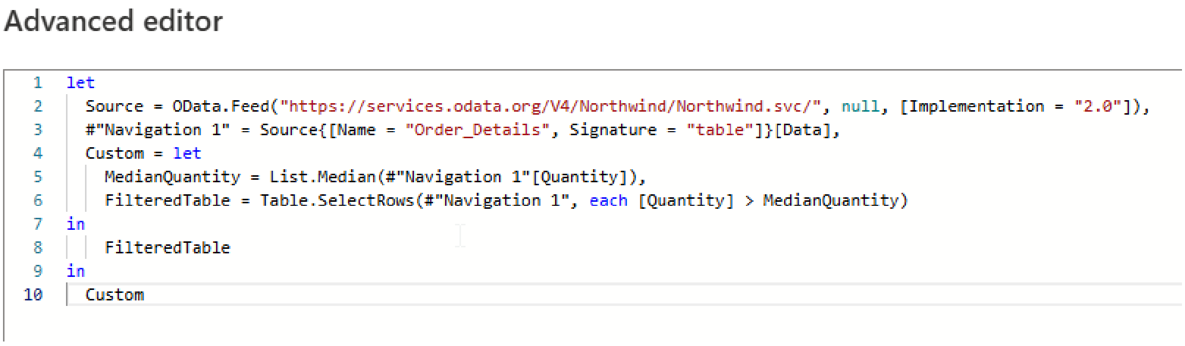


	OrderID	ProductID	UnitPrice	Quantity
1	10249	51	42.4	40
2	10250	51	42.4	35
3	10252	20	64.8	40
4	10252	33	2	25
5	10252	60	27.2	40
6	10253	39	14.4	42
7	10253	49	16	40
8	10254	55	19.2	21
9	10254	74	8	21
10	10255	16	13.9	35
11	10255	36	15.2	25
12	10255	59	44	30
13	10257	27	35.1	25
14	10258	2	15.2	50
15	10258	5	17	65
16	10260	57	15.6	50
17	10260	70	12	21
18	10263	16	13.9	60
19	10263	24	3.6	28
20	10263	30	20.7	60
21	10263	74	8	36
22	10264	2	15.2	35
23	10264	41	7.7	25
24	10265	17	31.2	30

11. On the **Power Query** toolbar, on the **Home** tab, select **Advanced editor**.

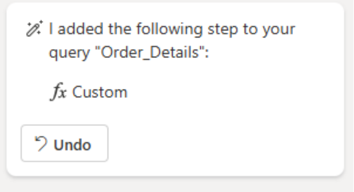


12. Review the definition of the formula used in the query.

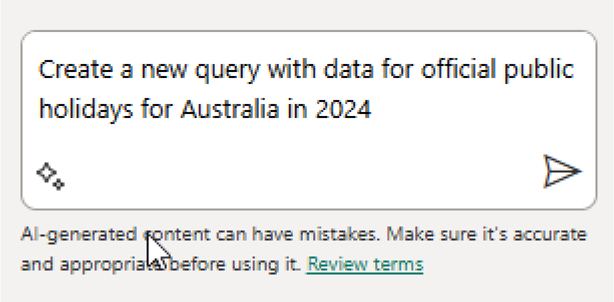

Advanced editor

```
let
  Source = OData.Feed("https://services.odata.org/V4/Northwind/Northwind.svc/", null, [Implementation = "2.0"]),
  #"Navigation 1" = Source{[Name = "Order_Details", Signature = "table"]}[Data],
  Custom = let
    MedianQuantity = List.Median(#"Navigation 1"[Quantity]),
    FilteredTable = Table.SelectRows(#"Navigation 1", each [Quantity] > MedianQuantity)
in
    FilteredTable
in
  Custom
```

13. Select **Cancel** to exit the Advanced editor without making changes.
14. In the **Copilot** pane, select **Undo** to revert the changes.


I added the following step to your query "Order_Details":

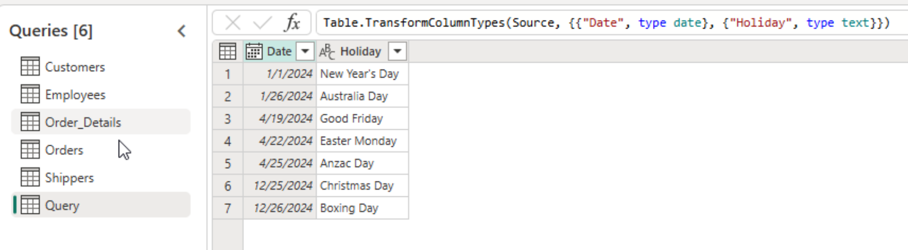
15. In the **Copilot** pane, enter *Create a new query with data for official public holidays for Australia in 2024*, then select **Send**.



Create a new query with data for official public holidays for Australia in 2024

AI-generated content can have mistakes. Make sure it's accurate and appropriate before using it. Review terms

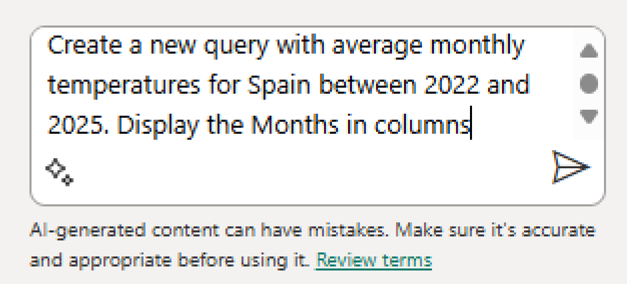
16. The Australian public holidays are now listed.



```
Table.TransformColumnTypes(Source, {{"Date", type date}, {"Holiday", type text}})
```

	Date	Holiday
1	1/1/2024	New Year's Day
2	1/26/2024	Australia Day
3	4/19/2024	Good Friday
4	4/22/2024	Easter Monday
5	4/25/2024	Anzac Day
6	12/25/2024	Christmas Day
7	12/26/2024	Boxing Day

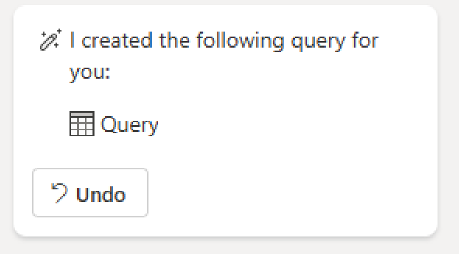
17. In the **Copilot** pane, select **Undo** to revert the changes.
18. In the **Copilot** pane, enter *Create a new query with average monthly temperatures for Spain between 2022 and 2025. Display the Months in columns* then select **Send**.



Create a new query with average monthly temperatures for Spain between 2022 and 2025. Display the Months in columns

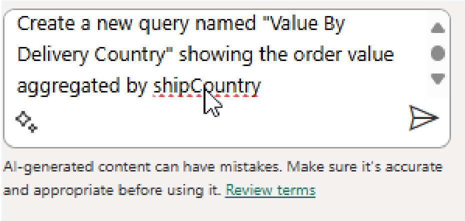
AI-generated content can have mistakes. Make sure it's accurate and appropriate before using it. Review terms

19. In the **Copilot** pane, select **Undo** to revert the changes.



I created the following query for you:

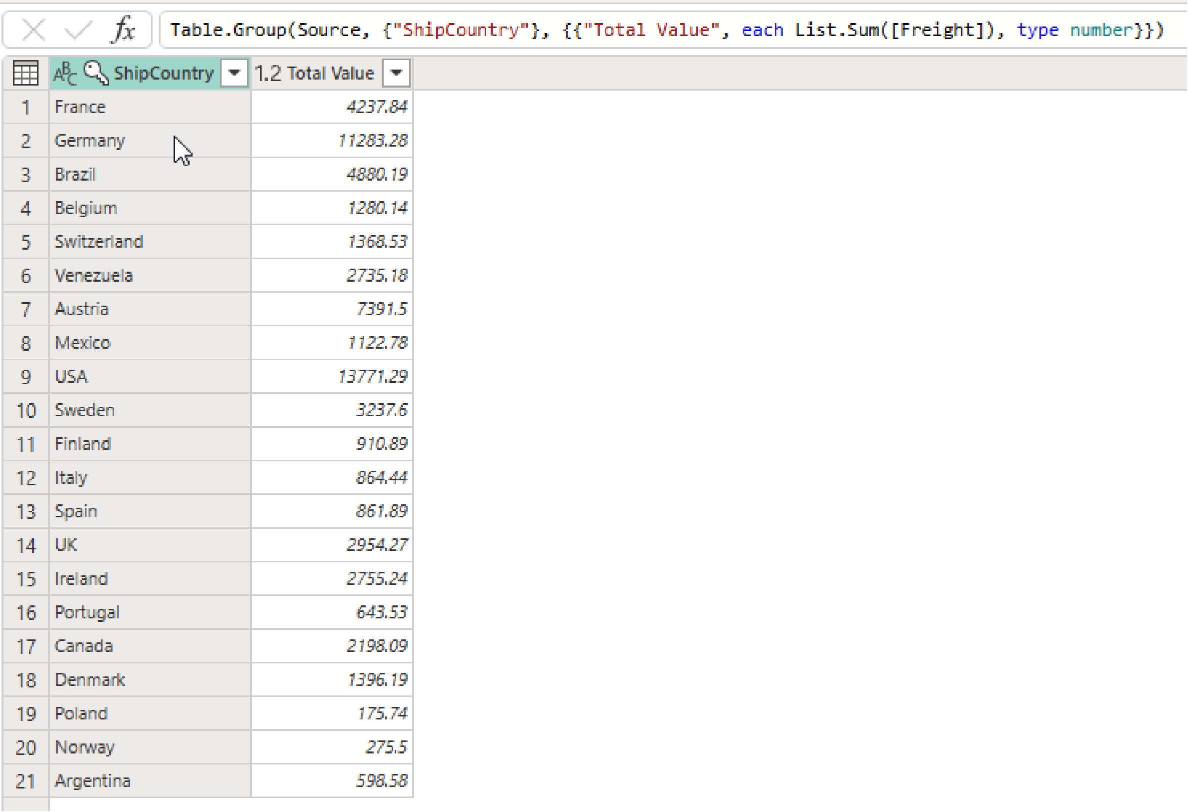
20. Select the **Orders** query.
21. In the **Copilot** pane, enter Create a new query named "Value By Delivery Country" showing the order value aggregated by shipCountry, then select **Send**.



Create a new query named "Value By Delivery Country" showing the order value aggregated by shipCountry

AI-generated content can have mistakes. Make sure it's accurate and appropriate before using it. Review terms

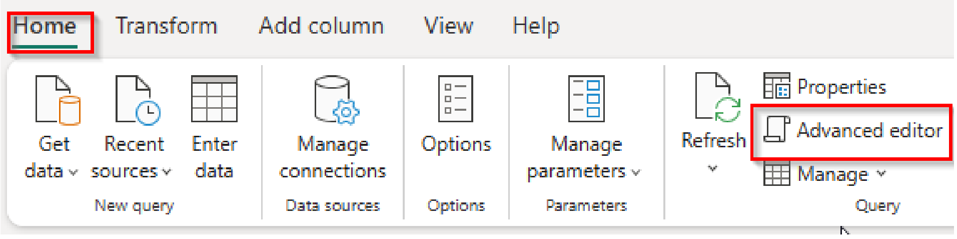
22. A table containing the shipCountry and order value aggregates is displayed.



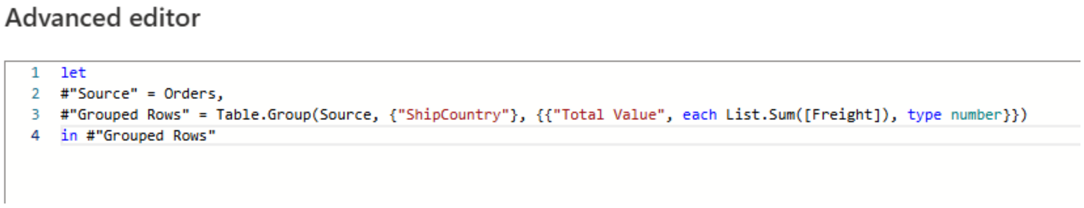
```
Table.Group(Source, {"ShipCountry"}, {{"Total Value", each List.Sum([Freight]), type number}})
```

	ShipCountry	1.2 Total Value
1	France	4237.84
2	Germany	11283.28
3	Brazil	4880.19
4	Belgium	1280.14
5	Switzerland	1368.53
6	Venezuela	2735.18
7	Austria	7391.5
8	Mexico	1122.78
9	USA	13771.29
10	Sweden	3237.6
11	Finland	910.89
12	Italy	864.44
13	Spain	861.89
14	UK	2954.27
15	Ireland	2755.24
16	Portugal	643.53
17	Canada	2198.09
18	Denmark	1396.19
19	Poland	175.74
20	Norway	275.5
21	Argentina	598.58

23. On the **Power Query** toolbar, on the **Home** tab, select **Advanced editor** to verify the correct formula was used.



24. The value of freight is being used, is that what we want? We need to check what Copilot is doing.



Advanced editor

```
let
#"Source" = Orders,
#"Grouped Rows" = Table.Group(Source, {"ShipCountry"}, {{"Total Value", each List.Sum([Freight]), type number}})
in #"Grouped Rows"
```

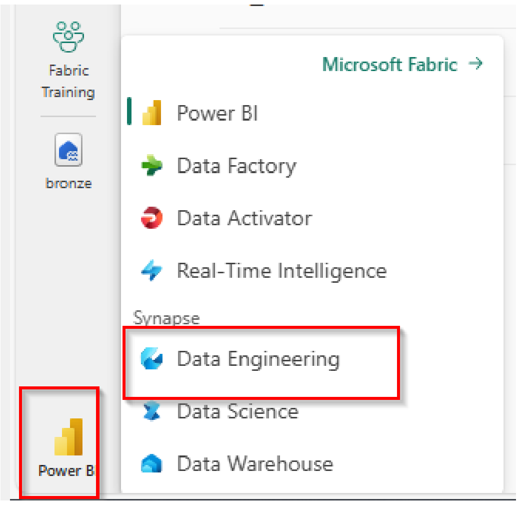
25. Select **Cancel** to close the Advanced editor, then in the **Copilot** pane, select **Undo** to revert the changes.
26. Select **Publish** to publish your data.

Exercise 2: Explore Copilot capabilities in Fabric notebooks

1. Open a new browser tab and go to https://app.powerbi.com/

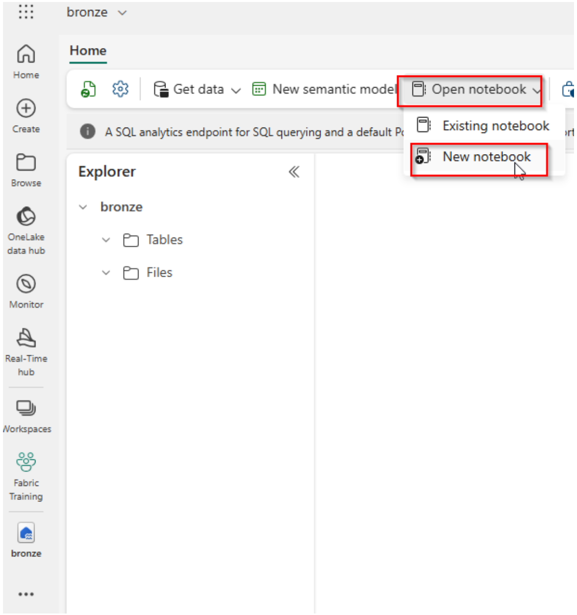
2. Open the Fabric training Workspace you have created previously.

3. In the lower left of the navigation pane for the workspace, select **Power BI**. Then, in the **Synapse** section, select **Data Engineering**.



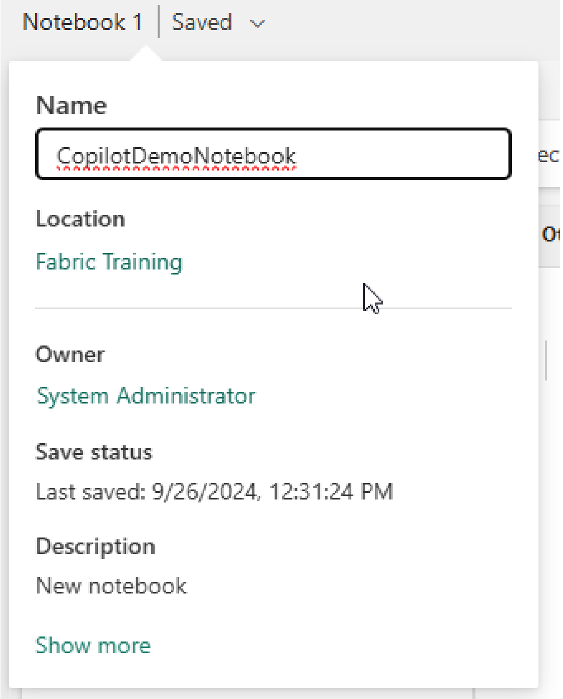
Select the bronze lakehouse you have created previously.

4. On the menu for the Bronze lakehouse, select **Open notebook** > **New notebook**.



A SQL analytics endpoint for SQL querying and a default P

5. At the upper left of the page, select the notebook name. Replace the name with **CopilotDemoNotebook** and select **Enter**



Name
CopilotDemoNotebook

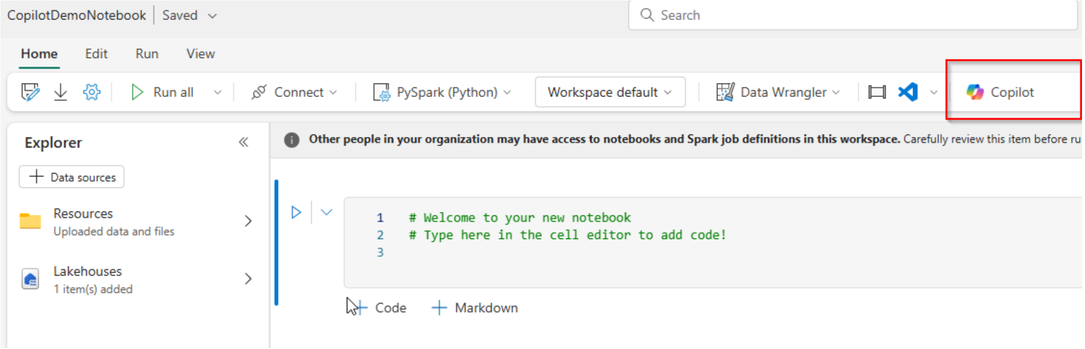
Location
Fabric Training

Owner
System Administrator

Save status
Last saved: 9/26/2024, 12:31:24 PM

Description
New notebook

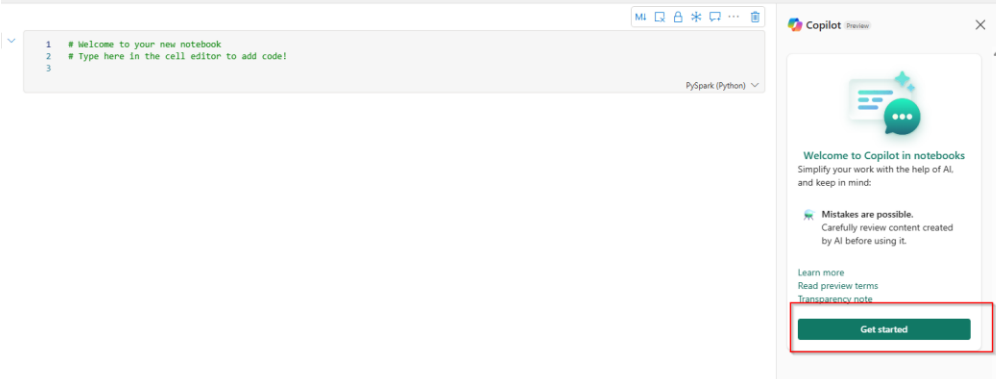
6. On the menu for the notebook, select **Copilot**.



Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before ru

```
# Welcome to your new notebook
# Type here in the cell editor to add code!
```

7. Select **Get Started**. Copilot adds a new cell to the notebook.

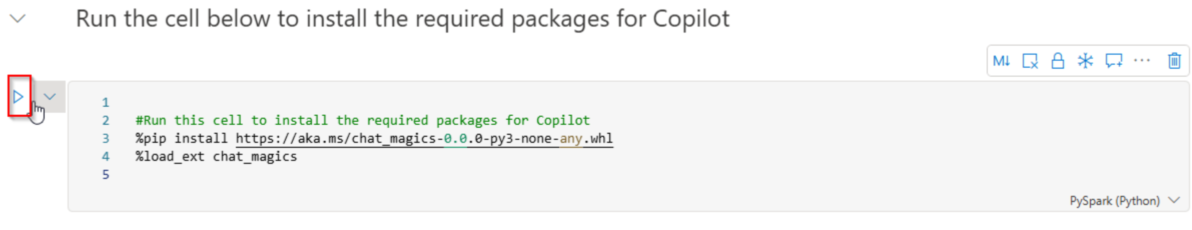


```
# Welcome to your new notebook
# Type here in the cell editor to add code!
```

Welcome to Copilot in notebooks
Simplify your work with the help of AI, and keep in mind:

Mistakes are possible.
Carefully review content created by AI before using it.

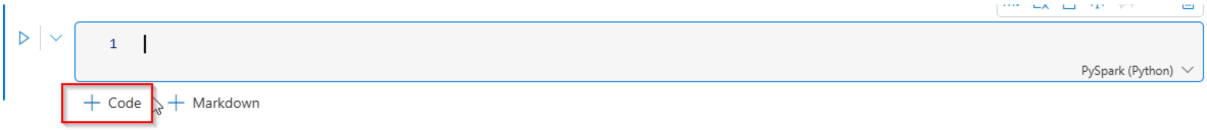
8. Select the **Run cell** button to install the packages that Copilot needs.



Run the cell below to install the required packages for Copilot

```
#Run this cell to install the required packages for Copilot
%pip install https://aka.ms/chat_magics-0.0.0-py3-none-any.whl
%load_ext chat_magics
```

9. Move the cursor to the lower left of the last cell in the notebook and select **+ code** to add a new cell.



10. Enter the following code in the new cell and then select **Run cell**.

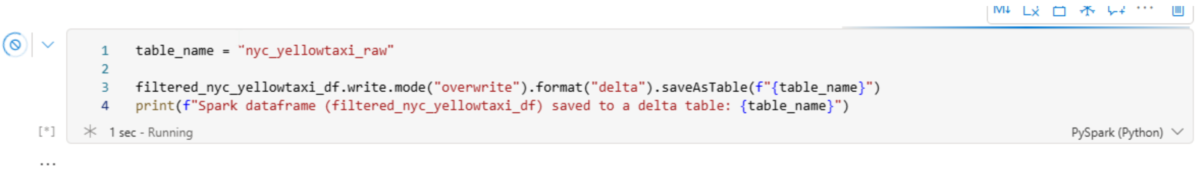
Note: This code specifies Azure storage access and connectivity information for the NYC Yellow Taxi open dataset. The last line of code filters the data to limit the volume of data that you'll ingest for this exercise

```
storage_account_name = "azureopendatastorage"
container_name = "nyctlc"

sas_token = r"" # Specify blank since container is public with anonymous access

spark.conf.set("fs.azure.sas.%s.%s.blob.core.windows.net" % (container_name, storage_account_name),sas_token)

directory = "yellow"
year = 2016
months = "1,2,3,4,5,6"
wasbs_path = f"wasbs://{container_name}@{storage_account_name}.blob.core.windows.net/{directory}"
nyc_yellowtaxi_df = spark.read.parquet(wasbs_path)

filtered_nyc_yellowtaxi_df = nyc_yellowtaxi_df.filter(f"puYear = {year} AND puMonth IN ({months})")
```

Warning: As each cell runs, you will see a message stating that Spark jobs are in progress. When processing completes you will see a messange stating that Spark jobs succeeded. If the code in a specific cell fails, processing for other cells does not run.

11. Add another cell to the notebook. Add the following code to the new cell and then select **Run cell** button. This code saves the data as a delta table in the Lakehouse.

```
table_name = "nyc_yellowtaxi_raw"

filtered_nyc_yellowtaxi_df.write.mode("overwrite").format("delta").saveAsTable(f"{table_name}")
print(f"Spark dataframe (filtered_nyc_yellowtaxi_df) saved to a delta table: {table_name}")
```



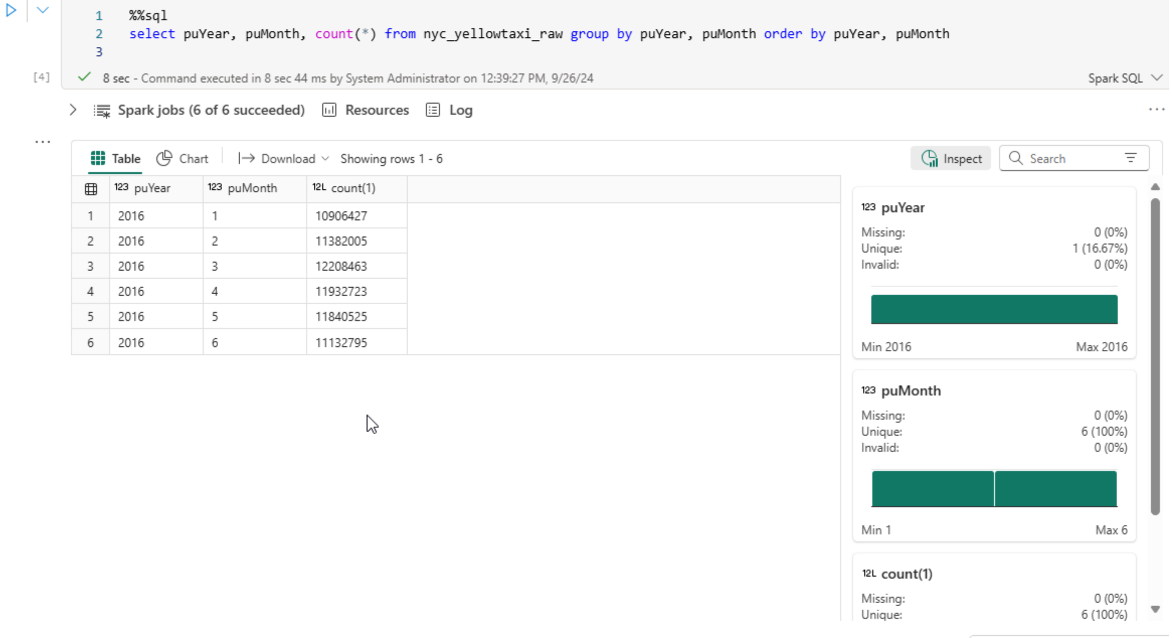
```
table_name = "nyc_yellowtaxi_raw"

filtered_nyc_yellowtaxi_df.write.mode("overwrite").format("delta").saveAsTable(f"{table_name}")
print(f"Spark dataframe (filtered_nyc_yellowtaxi_df) saved to a delta table: {table_name}")
```

[*] 1 sec - Running

12. Add another cell to the notebook. Add the following code to the new cell and then select **Run cell** button. This code runs a query to select and aggregate data.

```
%%sql
select puYear, puMonth, count(*) from nyc_yellowtaxi_raw group by puYear, puMonth order by puYear, puMonth
```

Note: You should see 6 rows in the ouput dataset from the query. Each row includes the year, month, and the number of records for that period.



```
%%sql
select puYear, puMonth, count(*) from nyc_yellowtaxi_raw group by puYear, puMonth order by puYear, puMonth
```

[4] 8 sec - Command executed in 8 sec 44 ms by System Administrator on 12:39:27 PM, 9/26/24

Spark jobs (6 of 6 succeeded)

Showing rows 1 - 6

	puYear	puMonth	count(1)
1	2016	1	10906427
2	2016	2	11382005
3	2016	3	12208463
4	2016	4	11932723
5	2016	5	11840525
6	2016	6	11132795

puYear
Missing: 0 (0%)
Unique: 1 (16.67%)
Invalid: 0 (0%)
Min 2016 Max 2016

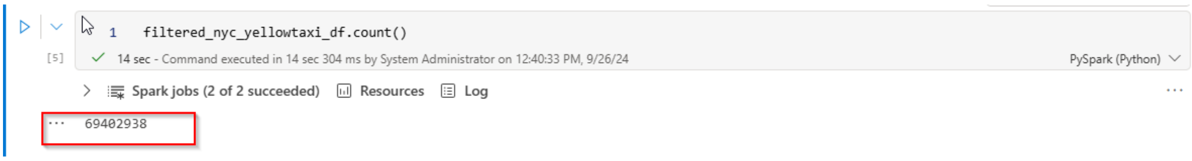
puMonth
Missing: 0 (0%)
Unique: 6 (100%)
Invalid: 0 (0%)
Min 1 Max 6

count(1)
Missing: 0 (0%)
Unique: 6 (100%)

13. Add another cell to the notebook. Add the following code to the new cell and then select **Run cell** button. This code counts the number of records returned.

```
filtered_nyc_yellowtaxi_df.count()
```

During testing, 69,402,938 rows were returned.



```
filtered_nyc_yellowtaxi_df.count()
```

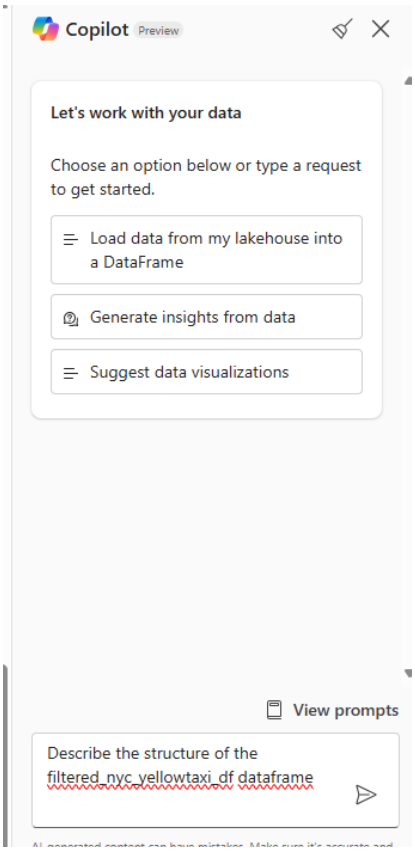
[5] 14 sec - Command executed in 14 sec 304 ms by System Administrator on 12:40:33 PM, 9/26/24

Spark jobs (2 of 2 succeeded)

```
69402938
```

14. At the bottom of the Copilot pane, enter the following prompt and then press the Enter key:

```
Describe the structure of the filtered_nyc_yellowtaxi_df dataframe
```



Let's work with your data

Choose an option below or type a request to get started.

Load data from my lakehouse into a DataFrame
Generate insights from data
Suggest data visualizations
View prompts

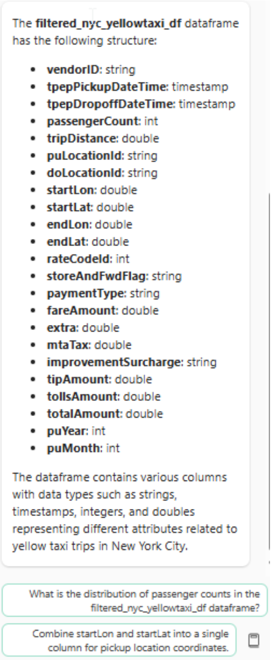
Describe the structure of the filtered_nyc_yellowtaxi_df dataframe

AI-generated content can have mistakes. Make sure it's accurate and

Warning: Copilot for Fabfric notebooks is in preview. During lab testing, we experienced mixed results when we submitted this prompt. In some cases, Coplilot responds with a Python command that you can enter in a cell to describe the dataframe structure. The command should resemenble the following:
filtered_nyc_yellowtaxi_df.describe().show()

In other cases, Copilot responded with "I'm unable to provide a description without more context or the structure of the data set." or "I must decline to assist with that request."

These issues should be resolved as this Copilot evolves.



The **filtered_nyc_yellowtaxi_df** dataframe has the following structure:

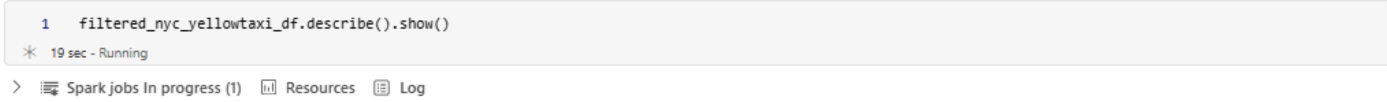
- **vendorID**: string
- **tpepPickupDateTime**: timestamp
- **tpepDropoffDateTime**: timestamp
- **passengerCount**: int
- **tripDistance**: double
- **puLocationId**: string
- **doLocationId**: string
- **startLon**: double
- **startLat**: double
- **endLon**: double
- **endLat**: double
- **rateCodeId**: int
- **storeAndFwdFlag**: string
- **paymentType**: string
- **fareAmount**: double
- **extra**: double
- **mtaTax**: double
- **improvementSurcharge**: string
- **tipAmount**: double
- **tollsAmount**: double
- **totalAmount**: double
- **puYear**: int
- **puMonth**: int

The dataframe contains various columns with data types such as strings, timestamps, integers, and doubles representing different attributes related to yellow taxi trips in New York City.

What is the distribution of passenger counts in the filtered_nyc_yellowtaxi_df dataframe?
Combine startLon and startLat into a single column for pickup location coordinates.

15. If Copilot does not create a command for you, add a new cell to the notebook. Then, add the following code to the new cell and then select **Run cell** button:

```
filtered_nyc_yellowtaxi_df.describe().show()
```

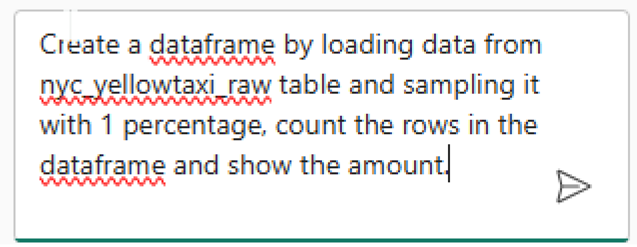


```
filtered_nyc_yellowtaxi_df.describe().show()
```

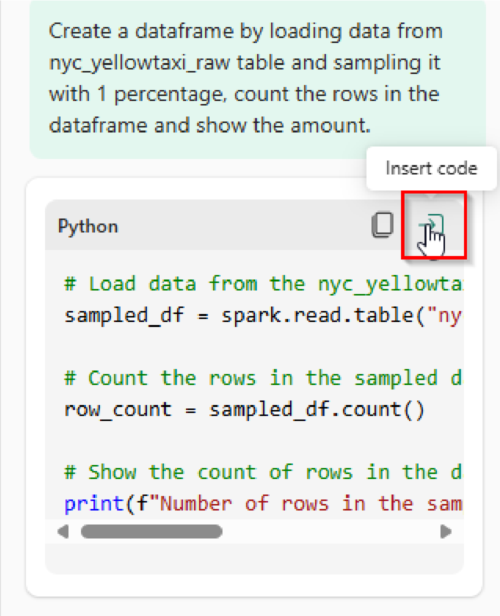
19 sec - Running

16. At the bottom of the Copilot pane, enter the following prompt and then press the Enter key. Copilot should respond with a command that you can run to create the dataframe.

Create a dataframe by loading data from nyc_yellowtaxi_raw table and sampling it with 1 percentage, count the rows in the dataframe and show the amount.



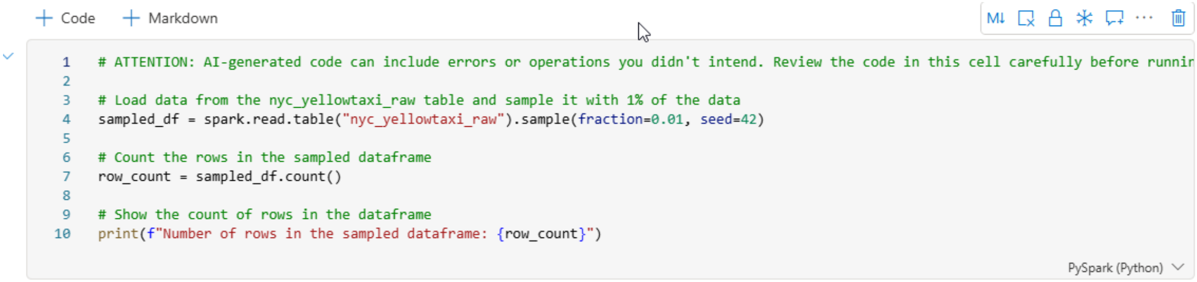
Create a dataframe by loading data from nyc_yellowtaxi_raw table and sampling it with 1 percentage, count the rows in the dataframe and show the amount.



Create a dataframe by loading data from nyc_yellowtaxi_raw table and sampling it with 1 percentage, count the rows in the dataframe and show the amount.

```
# Load data from the nyc_yellowta
sampled_df = spark.read.table("ny

# Count the rows in the sampled d
row_count = sampled_df.count()

# Show the count of rows in the d
print(f"Number of rows in the sam
```

17. Once you clicked on Code inserted, a new cell in the Notebook will be created. Execute the cell:



```
# ATTENTION: AI-generated code can include errors or operations you didn't intend. Review the code in this cell carefully before runnir

# Load data from the nyc_yellowtaxi_raw table and sample it with 1% of the data
sampled_df = spark.read.table("nyc_yellowtaxi_raw").sample(fraction=0.01, seed=42)

# Count the rows in the sampled dataframe
row_count = sampled_df.count()

# Show the count of rows in the dataframe
print(f"Number of rows in the sampled dataframe: {row_count}")
```

18. If Copilot does not create the command for you, add a new cell to the notebook. Then, add the following code to the new cell and then select **Run cell** button

```
%%code
Create a dataframe by loading data from nyc_yellowtaxi_raw table and sampling it with 1
percentage, count the rows in the dataframe and show the amount.
```

Important:
If you want to learn more about Chat-Magics, go to **Overview of chat-magics in Microsoft Fabric notebook**.